Identificación de la Fuente de Adquisición de Ficheros Multimedia de Dispositivos Móviles mediante Deep Learning

TRABAJO FIN DE GRADO

Doble Grado En Matemáticas e Ingeniería Informática

CURSO 2018–2019

Almudena Outeda Rodríguez

Directores

Luis Javier García Villalba

Ana Lucila Sandoval Orozco

Departamento de Ingeniería del Software e Inteligencia Artificial
Facultad de Informática
Universidad Complutense de Madrid

Madrid, Junio de 2019

Agradecimientos

Quisiera agradecer y dedicar este trabajo al Grupo de Investigación de Análisis, Seguridad y Sistemas (GASS), especialmente a mis tutores Luis Javier García Villalba y Ana Lucila Sandoval Orozco por haberme brindado la oportunidad de trabajar en un proyecto de tal magnitud, guiarme y darme facilidades para realizar un trabajo del que me siento orgullosa. A todos los miembros del grupo, especialmente a Esteban Alejandro Armas Vega por haber estado ahí en cada duda que me surgiera. A Jenny Cifuentes por las charlas de redes neuronales. Gracias a todos por guiarme y por toda la paciencia demostrada.

También quiero agradecer a mi familia y a mi pareja la paciencia que han tenido conmigo estos meses y la ayuda que me han brindado para poder realizar este trabajo.

Índice General

Índice de Figuras ix

Índice de Tablas xi

Lista de Acrónimos xiv

Abstract xvii

Resumen xix

1. Introducción 1

1.1. Motivación . . . 1
1.2. Contexto . . . 2
1.3. Objeto de la Investigación . . . 2
1.4. Trabajos Relacionados . . . 3
1.5. Plan de Trabajo . . . 3
1.6. Estructura del Trabajo . . . 4

2. Marco Teórico 7

2.1. Aprendizaje Automático . . . 7
2.1.1. Definición . . . 7
2.1.2. Conceptos, Metodologías y Modelos . . . 9
2.2. El Aprendizaje Profundo Frente al Aprendizaje Automático . . . 10
2.2.1. Representaciones de Datos y Reconocimiento de Características . . . 10
2.2.2. Arquitectura del Deep Learning y Estructura en Grandes Datos . . . 11
2.3. Las Redes Neuronales . . . 11
2.3.1. Redes Neuronales Artificiales . . . 11
2.3.2. Redes Neuronales Convolucionales . . . 13

3. Análisis Forense de Identificación de la Fuente de Adquisición 17

3.1. Originalidad del Contenido que Presenta el Fichero Multimedia . . . 18
3.1.1. Falsificación de Imágenes Digital . . . 18
3.1.2. Esteganografía Digital de Imágenes . . . 21

3.1.3. Falsificación de Vídeos 22
3.2. Identificación de la Fuente de Adquisición 23
3.2.1. Técnicas de Identificación de la Fuente de Adquisición en Imágenes . 24
3.2.2. Técnicas de Identificación de la Fuente de Adquisición en Vídeos . . 25
3.2.3. Identificación de la Fuente de Adquisición con Redes Neuronales Convolucionales 26

4. Estado del Arte **27**
4.1. Tipos de Enfoque en las Investigaciones 27
4.1.1. Basado en Modelos 27
4.1.2. Basado en Datos 28
4.2. Uso de las Redes Neuronales Convolucionales 28
4.2.1. Identificación de los Modelos de Cámara Tradicionales 29
4.2.2. Identificación de Cámaras de Móviles 30
4.2.3. Identificación de la Fuente de Adquisición de Vídeos 32

5. Algoritmo Propuesto **35**
5.1. Descripción del Algoritmo 35
5.1.1. Extracción de los Fotogramas de un Vídeo 35
5.1.2. Reducción de los Fotogramas en Áreas más Pequeñas 36
5.1.3. Preparación de los Datos 36
5.1.4. Clasificación de los *Patches* con la CNN Propuesta 36
5.1.5. Cálculo de la Precisión de la CNN con Vídeos a partir de la Precisión Obtenida con los Fotogramas 37
5.2. Desarrollo de la Red Propuesta 38
5.2.1. Primera Red Neuronal Convolucional 38
5.2.2. Segunda Red Neuronal Convolucional 39
5.3. Recursos y Herramientas Utilizadas 41
5.3.1. Configuración y Componentes de los Experimentos 41

6. Experimentos y Resultados **43**
6.1. Elección de los Conjuntos de Muestras 43
6.1.1. Conjuntos de Imágenes 43
6.1.2. Dataset Móviles GASS 44
6.1.3. Conjuntos de Vídeos 45
6.2. Tratamiento y Preprocesamiento de los Conjuntos 45
6.2.1. Selección y Creación de los Conjuntos 45
6.2.2. Separación en Dos Conjuntos: Entrenamiento y Validación 46
6.2.3. Extracción de los Fotogramas de cada Vídeo 46
6.2.4. División de cada Imagen en Áreas más Pequeñas 46
6.3. Resultados Obtenidos con la Red 1 47

6.3.1. Resultados del Conjunto 1 con la Red 1 47
6.3.2. Resultados del Conjunto 2 con la Red 1 48
6.4. Resultados Obtenidos con la Red 2 48
6.4.1. Resultados del Conjunto 1 con la Red 2 49
6.4.2. Resultados del Conjunto 2 con la Red 2 49
6.4.3. Resultados del Conjunto 3 con la Red 2 50
6.4.4. Resultados del Conjunto 4 con la Red 2 51

7. Conclusiones y Trabajo Futuro **53**
7.1. Conclusiones 53
7.2. Trabajo Futuro 54

8. Introduction **57**
8.1. Motivation 57
8.2. Context 58
8.3. Object of the Investigation 58
8.4. Related Works 59
8.5. Workplan 59
8.6. Struture of the Work 60

9. Conclusions and Future Work **63**
9.1. Conclusions 63
9.2. Future Work 64

Bibliografía **67**

Índice de Figuras

2.1. Historia de la Inteligencia Artificial y el Machine Learning 8
2.2. Pasos a seguir para desarrollar un modelo basado en Machine Learning . . . 9
2.3. Esquema de una red neuronal prealimentada. 12
2.4. Esquema de una red neuronal recurrente. 12
2.5. Esquema de la arquitectura de una red neuronal convolucional. En azul se muestran las capas convolucionales, en rojo, la capa *flattened* y en verde, las capas completamente conectadas. 14

3.1. Ejemplo del efecto Copy Paste [RTD11]. 18
3.2. Ejemplo del efecto de eliminar un objeto de una imagen [MA13]. 19
3.3. Ejemplo del efecto de retocar una imagen [RTD11]. 19
3.4. Ejemplo del efecto de realizar un montaje a partir de dos fotografías [BM13]. 19
3.5. Ejemplo de una imagen con esteganografía [MA13] 22

5.1. Esquema de la arquitectura de la Red 1. 39
5.2. Esquema de la arquitectura propuesta para la Red 2. 41

Índice de Tablas

4.1. Detalle de la red propuesta por [BBBT16] 30
4.2. Detalle de la red propuesta por [TCC16] 31

5.1. Detalles de la primera capa convolucional de la Red 1 38
5.2. Detalles de la segunda capa convolucional de la Red 1 38
5.3. Detalles de la tercera capa convolucional de la Red 1 39
5.4. Detalles de las capas completamente conectadas de la Red 1 39
5.5. Detalles de las capas convolucionales de la Red 2 40
5.6. Detalles de las capas completamente conectadas de la Red 2 40

6.1. Información sobre el Conjunto 1 44
6.2. Información sobre el Conjunto 2 y 3 45
6.3. Información sobre el Conjunto de Vídeos 45
6.4. Información sobre el Conjunto 4 46
6.5. Resultados de la Red 1 con el Conjunto 1 47
6.6. Resultados de la Red 1 con el Conjunto 1 48
6.7. Resultados de la Red 1 con el Conjunto 2 48
6.8. Resultados de la Red 2 con el Conjunto 1 49
6.9. Resultados de la Red 2 con el Conjunto 2 49
6.10. Resultados de la Red 2 con el Conjunto 2 50
6.11. Resultados de la Red 2 con el Conjunto 3 50
6.12. Resultados de la Red 2 con el Conjunto 4 52
6.13. Matriz de confusión del Conjunto 4 con la Red 2 52

Lista de Acrónimos

ANN *Artificial Neural Networks*

CFA *Color Filter Array*

CNN *Convolutional Neural Networks*

CPF *Conditional Probability Features*

DIT *Digital Image Tampering*

DSC *Digital Still Camera*

EBL *Explanation Based Learning*

FPN *Fixed Pattern Noise*

IQM *Image Quality Metric*

PCE *Peak to Correlation Energy*

PRNU *Photo-Response Non-Uniformity*

ReLU *Rectified Lineal Unit*

SPN *Sensor Pattern Noise*

SVM *Support-Vector Machine*

Abstract

Nowadays, society lives surrounded by multimedia content such as images and videos. The presence of electronic devices capable of taking photographs or recording videos is a reality in our daily lives, and their number increases with the passage of time. The vast majority of society carries a mobile phone in their pocket and makes use of it to take photos or videos. Associated to this, over the years have been appearing falsification techniques and manipulation of multimedia content that make it difficult to know if that content is authentic or not and where it comes from, which makes forensic analysis techniques a current necessity. In this work a convolutional neuronal network capable of identifying the source of acquisition of videos recorded with a mobile device is proposed.

The results that have been obtained in the experiments of this work shown clearly the effectiveness of the proposed methods. For the evaluation of these proposed methods there have been carried out experiments using a public dataset, which has been amplified with the literature and a generated dataset.

Keywords: Acquisition Source Identification, Classification, Digital Image, Digital Video, Forensics Analysis, Neuronal Convolution Network.

Resumen

Actualmente, la sociedad vive rodeada de contenido multimedia como son las imágenes y los vídeos. La presencia de dispositivos electrónicos capaces de realizar fotografías o grabar vídeos es una realidad en nuestra vida cotidiana, y su número aumenta con el paso del tiempo. La gran mayoría de la sociedad lleva un móvil en el bolsillo y hace uso de él para realizar fotos o vídeos. Ligado a ello, con los años han ido apareciendo técnicas de falsificación y manipulación de contenidos multimedia que dificultan saber si ese contenido es auténtico o no y de dónde procede, lo que hace que las técnicas de análisis forense sean una necesidad actual. En este trabajo se propone una red neuronal convolucional capaz de identificar la fuente de adquisición de vídeos grabados con un dispositivo móvil.

Los resultados obtenidos de los experimentos realizados en este trabajo demuestran la eficiencia de métodos propuestos. Para la evaluación de los métodos propuestos se realizaron experimentos con un *dataset* público ampliamente utilizado en la literatura y un *dataset* generado.

Palabras clave: Análisis Forense, Clasificación, Identificación de Fuente, Imágenes Digitales, Red Neuronal Convolucional, Vídeos Digitales.

Capítulo 1

Introducción

1.1. Motivación

Hoy en día, la gran mayoría de la población lleva un teléfono móvil en el bolsillo, cuando no más de uno. En sus inicios, estos dispositivos tenían funciones muy básicas, como las de realizar o recibir llamadas o ser capaces de enviar y recibir mensajes. Poco a poco, se les ha ido añadiendo más funcionalidades como juegos, calendarios y cámaras móviles. A día de hoy, la inmensa mayoría de los móviles disponen de al menos, una cámara fotográfica, capaz de hacer tanto fotografías como vídeos. Esto, sumado a la sencillez con la que pueden usarse estos dispositivos, hace que el hecho de realizar una fotografía o vídeo en cualquier momento y en cualquier lugar sea un acto normalizado en el día a día de cualquier persona. Con frecuencia, hasta hace unos años, los contenidos que pueden verse en una fotografía o vídeo han sido considerados realidades incuestionables. Sin embargo, se ha producido un auge de las tecnologías y métodos de falsificación y alteración de imágenes y vídeos y de su uso, incluso entre aquellas personas que no tienen conocimientos en este campo [GKWB07]. Este fenómeno ha implicado que ya no se pueda confiar en lo que representa una imagen ni un vídeo nunca más. Por ello, a la vez que el número de engaños en las imágenes aumenta, se ha desarrollado paralelamente una serie de estudios e investigaciones que sean capaces de determinar de manera unívoca si una imagen o un vídeo ha sido alterado o muestra su contenido original.

Uno de los enfoques más importantes que se le ha dado a estas investigaciones y avances es el intento de descubrir qué cámara fue la que grabó un determinado vídeo o imagen. Esto se hace a partir del propio vídeo o imagen, sin más ayuda que la información que pueda contener. Afortunadamente, a la hora de grabarse, cada cámara deja en los elementos capturados una huella propia con diferentes características, lo que permite hacer un seguimiento de esas características y diferenciar una cámara de otra.

Los avances en este campo son dispares según si se trata de imágenes o vídeos, o de si fueron grabados con una cámara tradicional o con la cámara de un dispositivo móvil, pero todos ellos son muy importantes para poder mejorar estas técnicas de identificación y ser útiles a la sociedad. Gracias a estos trabajos, es posible determinar si un determinado vídeo o fotografía puede ser usado como prueba judicial y confiar plenamente en el contenido que se muestra. Esto ha facilitado la admisión de imágenes y vídeos de diversas cámaras, especialmente las videocámaras de seguridad como pruebas judiciales, y han ayudado a la identificación de culpables a partir de la identificación del dispositivo que grabó las imágenes.

1.2. Contexto

El presente Trabajo Fin de Grado se enmarca dentro de un proyecto de investigación titulado RAMSES aprobado por la Comisión Europea dentro del Programa Marco de Investigación e Innovación Horizonte 2020 (Convocatoria H2020-FCT-2015, Acción de Innovación, Número de Propuesta: 700326) y en el que participa el Grupo GASS del Departamento de Ingeniería del Software e Inteligencia Artificial de la Facultad de Informática de la Universidad Complutense de Madrid (Grupo de Análisis, Seguridad y Sistemas, http://gass.ucm.es, grupo 910623 del catálogo de grupos de investigación reconocidos por la UCM).

Además de la Universidad Complutense de Madrid participan las siguientes entidades:

- Treelogic Telemática y Lógica Racional para la Empresa Europea SL (España)
- Ministério da Justiça (Portugal)
- University of Kent (Reino Unido)
- Centro Ricerche e Studi su Sicurezza e Criminalità (Italia)
- Fachhochschule fur Offentliche Verwaltung und Rechtspflege in Bayern (Alemania)
- Trilateral Research & Consulting LLP (Reino Unido)
- Politecnico di Milano (Italia)
- Service Public Federal Interieur (Bélgica)
- Universität des Saarlandes (Alemania)
- Dirección General de Policía - Ministerio del Interior (España)

1.3. Objeto de la Investigación

Se ha demostrado que es posible identificar la fuente de adquisición de una imagen o un vídeo a través de las características que presentan sus elementos internos y sus metadatos. Estas características, que son adquiridas durante su proceso de creación y en procesamientos posteriores, forman parte de lo que se podría considerar el ADN de una imagen o vídeo, y siendo analizadas, pueden mostrar información determinante sobre un contenido digital. Parte de estas señas de identidad son otorgadas por la cámara fuente que tomó la imagen o vídeo, que deja una huella propia sobre sus creaciones, y que permitirá identificarla a posteriori.

La mayoría de las investigaciones realizadas en los últimos años sobre técnicas de identificación de fuente se han enfocado únicamente en la identificación de cámaras tradicionales *Digital Still Camera* (DSC). Con la aparición y masificación de los dispositivos móviles, en su mayoría capaces de capturar fotografías, se consideró necesario realizar nuevas investigaciones sobre las técnicas para identificar la fuente de imágenes generadas por dispositivos móviles. La gran mayoría de estos estudios se han realizado enfocándose en imágenes, habiendo muy pocos sobre vídeos.

La presente investigación se centra en las técnicas de identificación de fuente de un vídeo, ya que es un campo mucho menos estudiado. También, se tratarán las técnicas de identificación de la fuente de imágenes, ya que tienen una relación importante. Este trabajo propone una red neuronal convolucional capaz de identificar el dispositivo móvil que generó un vídeo a partir de sus fotogramas.

1.4. Trabajos Relacionados

Las tareas de análisis forense de imágenes digitales pueden dividirse en cuatro ramas, en función del objetivo que persigan [CFGL08]:

- Verificación de integridad.
- Recuperación de la historia de procesamiento.
- Clasificación basada en la fuente.
- Agrupación por dispositivo fuente e identificación de la fuente.

Dentro de los cuatro grupos, el *modus operandi* que se utiliza es muy similar: con el objetivo de implementar un algoritmo que ayude al análisis o desarrollar un método nuevo que facilite la clasificación, la comunidad científica estudia las características que componen o contiene un contenido digital, para poder usar la información de las mismas en sus investigaciones. Estas características, que pueden ser de muy distinta naturaleza, como las peculiaridades del sensor de la cámara o la distorsión de la lente, son la piedra angular del análisis forense. En [VLCEK07, TNC10] se lleva a cabo un estudio sobre cuáles pueden ser las características de una imagen que pueden ser interesantes o útiles para el análisis forense en dispositivos móviles.

En la literatura disponible se pueden encontrar diversos trabajos basados en el uso de una o varias de estas características, de los cuales destacan principalmente [BBBT16, TCC16, FONBCS18, BLG+17, MG18, YHW12]. Su punto común y el motivo de ser especialmente importantes para este trabajo es la utilización por parte de casi todos de una red neuronal convolucional para la identificación de la fuente de adquisición, siendo la misma técnica que se usará en este trabajo. [YHW12] no hace uso de una *Convolutional Neural Networks* (CNN), pero al tratarse de una investigación sobre la fuente de adquisición de un vídeo, al igual que este trabajo, se ha visto interesante añadir sus conclusiones. Cabe destacar que la literatura disponible sobre el análisis de la identificación de las fuentes de adquisición es muy desigual según si tratan sobre imágenes o vídeos. Sobre las primeras existe un número aceptable de publicaciones, aunque no muy numeroso. En cambio, el número de trabajos que se centran en la identificación de la fuente de adquisición de un vídeo es muy inferior. Por ello, animados por los resultados obtenidos en la literatura al aplicar las técnicas de Machine Learning a las imágenes para clasificarlas según la cámara que las capturó, en este trabajo se propone aplicar esas mismas técnicas a los vídeos. Como resultado, la propuesta de este trabajo es el uso de una red CNN para su clasificación. Su elección se debe a ser una técnica muy poco estudiada en vídeos aún, y a que según los resultados de la literatura más actual, es uno de los métodos más rápidos y eficaces que existe hoy en día. Además, es un método capaz de adaptarse a distintos propósitos con facilidad.

1.5. Plan de Trabajo

El desarrollo de este trabajo se ha realizado en tres fases:

1. **Investigación**: Esta primera fase se llevó a cabo en un periodo aproximado de tres meses. Comenzó con la elección de la temática del trabajo junto con los tutores: la identificación de la fuente de adquisición de vídeos digitales. Una vez elegido se inició la investigación sobre la temática del proyecto en el que se enmarca el trabajo. Era necesario saber qué se había hecho ya al respecto para poder avanzar

y ampliar los resultados ya conocidos. Gracias principalmente a las asignaturas de Geometría Computacional e Inteligencia Artificial, se partía con una pequeña base de conocimientos sobre Aprendizaje Automático, Aprendizaje Profundo y Redes Neuronales, de los que se hablará en los próximos capítulos. Estos conocimientos fueron rápidamente ampliados gracias a los numerosos artículos de revistas científicas, de congresos y algunos libros sobre la materia, todos ellos encontrados tanto en *Google Scholar* como en las bibliotecas de la Universidad Complutense de Madrid. Se buscaron principalmente publicaciones sobre Aprendizaje Automático, Redes Neuronales (especialmente las convolucionales), falsificación de imágenes y vídeos e identificación de imágenes y vídeos. Muchos de los trabajos utilizados en la fase de investigación fueron encontrados gracias a las referencias de otras publicaciones ya estudiadas.

2. **Desarrollo**: Una vez que la fase de investigación estuvo suficientemente avanzada y se adquirieron todos los conocimientos necesarios, se comenzó la segunda fase, la del desarrollo de la propuesta. En este momento se determinó cuál sería la propuesta que presentaría este trabajo: el desarrollo de un método que permita identificar la fuente de adquisición de un vídeo a partir de una CNN. A la vez que se desarrolló y codificó la propuesta, se siguió también con la investigación, aunque en menor medida que en la fase anterior, solo buscando y consultando conceptos puntuales. En esta fase se llevó a cabo la elección y creación de los conjuntos de imágenes y vídeos que serían utilizados para entrenar y evaluar la red propuesta y se desarrolló, codificó, probó y rediseñó la red en numerosas ocasiones hasta lograr el diseño definitivo. Para ello, se investigó sobre el lenguaje de programación *Python* y sus librerías *Keras, Tensorflow* y *Scikit Learn*.

3. **Resultados**: Finalmente, en la fase de Resultados se procedió al análisis de todos los resultados que se habían obtenido con las numerosas pruebas realizadas y a la extracción de conclusiones a partir de dichos resultados. Esta fase abarcó en torno a los dos meses y fue solapada un mes y medio con la fase anterior, ya que según los resultados que se obtuvieran, era necesario realizar más pruebas o no.

1.6. Estructura del Trabajo

El resto del trabajo está organizado en 9 capítulos con la estructura que se explica a continuación:

En el Capítulo 2 se introducen los conceptos básicos del Aprendizaje Automático, el Aprendizaje Profundo y las Redes Neuronales, que son elementales para comprender las bases de la ciencia en las que está cimentado el análisis forense en imágenes y vídeos.

En el Capítulo 3 se explica qué es la identificación de la fuente de adquisición de una imagen o vídeo, qué técnicas forenses existen para ello y en qué elementos de las cámaras se basan para poder aplicarse. En este capítulo se comienza mostrando los diferentes objetivos que persigue el análisis forense: la determinación de si un contenido es original o no y la identificación de la cámara fuente. En ambos se realiza una presentación de los métodos y técnicas que utilizan todos ellos. Por último, se muestra de forma más detallada en qué consiste la identificación de la fuente de adquisición a partir de una CNN, pues es una parte principal de este trabajo.

El Capítulo 4 recoge una recopilación de las investigaciones más importantes de la identificación de la fuente de adquisición. En él, se pueden leer los logros obtenidos por [BBBT16, TCC16] en la identificación de la cámara fuente de imágenes captadas por una

cámara DSC tradicional y como [FONBCS18] lleva dichos resultados un paso más allá aplicando esas mismas técnicas a imágenes obtenidas a partir de dispositivos móviles. A continuación, se exponen los avances de [MG18, YHW12] en el terreno de los vídeos, tratando incluso aquellos que han sido transmitidos por una red social.

El Capítulo 5 contiene la propuesta de este trabajo. En él se explica de forma general en qué consiste dicha propuesta y a continuación se detalla la red final que se propone.

El Capítulo 6 describe los experimentos realizados para evaluar la efectividad de la red neuronal convolucional propuesta en el capítulo 5 y presenta los resultados obtenidos.

El Capítulo 7 muestra las principales conclusiones de este trabajo y las posibles líneas futuras de investigación.

Los Capítulos 8 y 9 son las traducciones al inglés de la Introducción (este capítulo) y de las Conclusiones (Capítulo 7).

Capítulo 2

Marco Teórico

El objetivo de este capítulo es explicar diferentes conceptos generales relacionados con el Aprendizaje Automático, el Aprendizaje Profundo y las redes neuronales, para facilitar al lector la comprensión de los siguientes capítulos.

2.1. Aprendizaje Automático

Este trabajo tiene una base muy fuerte en el Aprendizaje Automático, por lo que es necesario realizar una introducción a este concepto, y así facilitar la lectura y la comprensión al lector del trabajo.

2.1.1. Definición

El Aprendizaje Automático es un subcampo de las ciencias de la computación y una rama de la Inteligencia Artificial. Este área de conocimiento estudia el desarrollo de diversas técnicas para que las computadoras aprendan y sean capaces de llegar a una conclusión tras el estudio de cierta información previamente proporcionada. Es mayormente conocida por su nombre inglés: Machine Learning.

El Machine Learning nació con el *Test de Turing*, creado por Alan Turing en 1950. El objetivo de dicho test era determinar si una máquina era inteligente o no. Para lograrlo, debía ser capaz de hacer creer a un humano que era humana en vez de un computador [Kub17, McC04, Buc05]. A partir de aquí, se produjeron importantes avances en los siguientes años, los cuáles vienen representados en la Figura 2.1.

El primero fue el de Arthur Samuel, que desarrolló en 1952 un programa capaz de aprender por sí mismo. Se trataba de un software que jugaba a las damas y que mejoraba sus jugadas a cada partida que realizaba. Cuatro años más tarde, Martin Minsky y John McCarthy organizaron una conferencia en Darthmouth, donde se considera que nace el campo de Inteligencia Artificial. Finalmente, en 1958 Frank Rosenblatt diseña el Perceptrón, la primera red neuronal artificial.

Entre los años 1974 y 1980 se produjo un desinterés generalizado en la Inteligencia Artificial, lo que llevó a un estancamiento en su desarrollo conocido como Primer Invierno de la Inteligencia Artificial. Tras este parón, se produjo un nuevo interés en los años 80, conocido como la explosión de los 80. Estos años se caracterizan por el nacimiento de sistemas expertos, basados en reglas. En esta época, Gerald Dejong introduce el concepto de *Explanation Based Learning* o *Explanation Based Learning* (EBL). Éste consiste en que un computador sea capaz de analizar datos de entrenamiento y crear reglas generales, lo que le permite descartar datos irrelevantes. Más adelante, en 1985, Terry Sejnowski

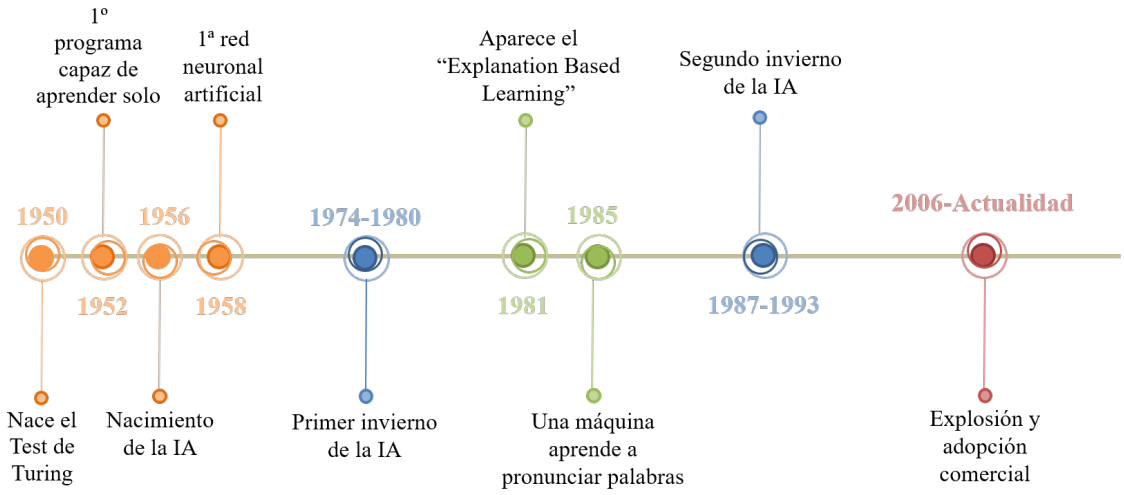

Figura 2.1: Historia de la Inteligencia Artificial y el Machine Learning

inventa una máquina capaz de aprender a pronunciar palabras como un niño, a la que bautiza como NetTalk.

Tras un Segundo Invierno de la Inteligencia Artificial entre los años 1987 y 1993, el Machine Learning comenzó a enfocarse en diferentes asuntos tales como el razonamiento probabilístico, la investigación basada en la estadística y la recuperación de información. Además de todas estas áreas, su objetivo principal y en el que continuó profundizando sobre todo y cada vez más fue el reconocimiento de patrones. Por ello, en la década de los 90 el Machine Learning se acabó separando por completo de la Inteligencia Artificial para convertirse en una disciplina autónoma. Hoy en día, su principal objetivo es abordar y resolver problemas prácticos en donde se aplique cualquiera de las disciplinas numéricas antes mencionadas.

El Aprendizaje Automático utiliza el software para detectar patrones y tendencias en datos, de manera que estos mejoren con el tiempo en el caso de añadir datos nuevos. Además, el Machine Learning se apoya también en una serie de técnicas estadísticas, ya que partes clave que lo componen se basan en el análisis estadístico y en el uso de modelos estadísticos para obtener predicciones basadas en patrones y tendencias.

A pesar de que habitualmente se asocie el Machine Learning con la Inteligencia Artificial, este proceso, más que ser inteligente, lo que hace es 'aprender' aquello que queramos enseñarle a identificar. Para ello, en vez de proporcionarle una definición (de lo que es un perro, por ejemplo), que no serviría de nada, se le facilitará un conjunto de ejemplos de imágenes dónde aparezcan perros.

Cuanto más grande y variado sea ese conjunto, más facilidad tendrá para identificar al perro en una imagen distinta a las habituales en un futuro. A este conjunto se le llamará *conjunto de entrenamiento*. Una vez el modelo ha sido creado, se usará otro conjunto, diferente del de entrenamiento, para evaluar la fiabilidad y precisión del modelo. Este será el *conjunto de validación*. Gracias al Machine Learning, la ciencia de los datos y el análisis ha podido avanzar en varios campos, como el de medicina o los fraudes, convirtiéndose en una herramienta clave en la respuesta de varios tipos de problemas, como:

1. ¿Es X o es Y?

2. ¿Cuál es el valor numérico más probable de Y para X?

3. ¿Es algo fuera de lo normal o inesperado?

4. ¿Cómo están estructurados los datos?

2.1.2. Conceptos, Metodologías y Modelos

Para llevar a cabo un proceso de análisis de datos usando Machine Learning y desarrollar un modelo en base a ello, es necesario seguir una serie de pasos, los cuáles muestra de forma esquemática la Figura 2.2 [CSA+18, RM15, JM15].

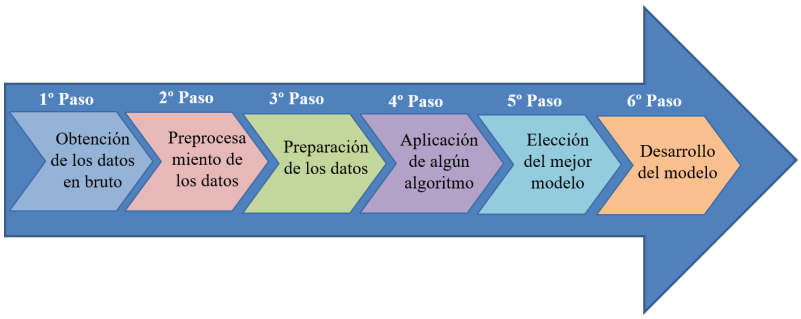

Figura 2.2: Pasos a seguir para desarrollar un modelo basado en Machine Learning

1. **Obtención de datos en bruto.** Los datos a evaluar pueden proceder de cualquier fuente y estar ya estructurados o no.

2. **Preprocesamiento de los datos.** Para poder trabajar con ellos, es necesario prepararlos, extrayendo las características interesantes o bien eliminando datos innecesarios o incompletos.

3. **Preparación de los datos.** Antes de proporcionárselos al modelo, habrá que añadir a los datos etiquetas o cambiar el formato numérico.

4. **Aplicar uno o más algoritmos de Machine Learning.** Es normal tener que aplicar varios algoritmos y testear los modelos hasta descubrir cuál es el que mejor se adapta a un caso concreto.

5. **Determinar cuál es el mejor modelo.** Lo cual se consigue aplicando la lógica del mundo real o de los negocios.

6. **Desarrollar el modelo.** Esto proporciona un programa codificado que servirá para usarlo como usuario final.

Es común que lo que más tiempo abarque sea la preparación de los datos. Una vez generada la tabla con todos los datos preparados, cada columna de la misma será una característica, y se usará algún tipo de selector de características para identificar aquellas que son relevantes para el modelo.

Un modelo de Machine Learning es el código que se genera después de que un algoritmo se haya ejecutado con los datos, y los patrones u otros resultados hayan sido identificados. El código es, por lo tanto, el conjunto de instrucciones que resultaron exitosas en la resolución del problema planteado con el conjunto de datos de entrenamiento:

- **Entrenamiento de un Modelo**: Dado que es muy raro que el primer algoritmo usado funcione perfectamente a la primera, será necesario ir haciendo pruebas con distintos algoritmos a ver cuál de todos proporciona los mejores resultados. A este proceso se le llama *experimentación*. Durante el entrenamiento, un paso clave es la *evaluación del modelo*: Esto permite observar qué valores adquieren los resultados, ejecutando el modelo con datos diferentes de los usados durante el entrenamiento [Kub17, JM15, CSA+18].

- **Desarrollo de un Modelo**: Una vez identificado el mejor modelo, es necesario desarrollarlo para poder aplicarlo a cualquier conjunto de datos. El desarrollo es el último paso del proceso y sólo se lleva a cabo cuando el testeo del modelo ha dado resultados que merecen la pena. Sin embargo, en algunos casos se lleva a cabo un proceso de post-desarrollo, en el que después de desarrollar el modelo, se comprueba que sigue devolviendo resultados satisfactorios y que se adapta bien si se le añaden más cantidad de datos [Kub17, JM15, CSA+18].

2.2. El Aprendizaje Profundo Frente al Aprendizaje Automático

Como se ha podido ver en el apartado anterior, los sistemas basados en el Aprendizaje Automático o Machine Learning son usados en una gran variedad de situaciones para resolver problemas. A su vez, cada vez más, estas aplicaciones hacen uso de unas nuevas técnicas llamadas Aprendizaje Profundo o Deep Learning. El Deep Learning nace como una vertiente o subcampo del Machine Learning, dando un salto más allá e incorporando un avance significativo: para enseñar a la máquina a aprender a resolver un problema, se utilizan unas estructuras lógicas basadas en la organización del sistema nervioso de un mamífero.

Durante décadas se han ido mejorando las técnicas de Machine Learning. Las técnicas convencionales de Machine Learning fueron muy limitadas en su capacidad de procesar los datos de forma cruda. Esto provocaba que construir un patrón de reconocimiento de características necesitase una gran experiencia en el manejo de datos y un nivel muy alto de ingeniería. El Deep Learning ayudó a paliar ese problema y mejoró sensiblemente el análisis de datos [LBH15, RM15, Agg18].

2.2.1. Representaciones de Datos y Reconocimiento de Características

El aprendizaje por representación es una serie de métodos que permite a la máquina recibir datos sin procesar y descubrir automáticamente las muestras que necesita para clasificarlos. El Deep Learning permite a los modelos computacionales que están compuestos por múltiples capas de procesadores (del inglés, *multiple processing layers*) aprender representaciones de datos con múltiples niveles de abstracción.

Los métodos del Deep Learning son métodos de aprendizaje por representación con múltiples niveles de representación. Son obtenidos componiendo módulos simples y no lineales, en los que cada uno transforma la representación a un nivel en una representación de un nivel más preciso y abstracto.

Estas altas capas de representación son la clave a la hora de realizar tareas de reconocimiento y clasificación, ya que centran su atención en las características más relevantes a la hora de la discriminación de los datos de entrada, eliminando aquellas que no son importantes para clasificar. Es capaz de ser eficaz incluso en datos con

estructuras intrínsecas muy complejas, siendo especialmente valioso por ello. La clave del Deep Learning está en que estas capas de características no son diseñadas por un ingeniero humano: han sido debidamente aprendidas desde los datos tras un procedimiento de aprendizaje de propósito general.

Se cree que el Deep Learning tendrá mucha importancia en un futuro cercano, ya que sería capaz de gestionar de manera provechosa el aumento en la cantidad de cómputo y datos disponibles. Además, este proceso será acelerado con los algoritmos que se encuentran en desarrollo para redes neuronales [Agg18, LBH15].

2.2.2. Arquitectura del Deep Learning y Estructura en Grandes Datos

El Deep Learning basa su arquitectura en unas estructuras llamadas redes, las cuales hay de varios tipos. Las redes convolucionales profundas han traído avances en el procesamiento de imágenes, vídeo, voz y audio, mientras que las redes recurrentes han arrojado luz sobre los datos secuenciales, como texto y voz.

La arquitectura de estas redes del Deep Learning en una pila multicapa de módulos simples, dónde todos (o la gran mayoría) están sujetos a aprendizaje, y muchos de ellos computan asignaciones de entrada-salida no lineales. Cada módulo de la pila transforma su entrada con el fin de incrementar tanto la selectividad como la invariancia de la representación. Con múltiples capas no lineales (por ejemplo, con una profundidad entre 5 y 20), un sistema puede implementar funciones realmente complejas a partir de sus entradas, que son extremadamente sensibles a pequeños detalles de los datos.

Gracias a ello, el Deep Learning es capaz de descubrir estructuras muy complejas en grandes conjuntos de datos. Para ello, se vale de un algoritmo de retropropagación o algoritmo de *backpropagation* en inglés. Con él indica a la máquina cómo debería cambiar sus parámetros internos que ha usado para computar la representación de los datos en cada capa desde la representación de la capa anterior.

La finalidad de este ajuste es minimizar todo lo posible la función objetivo. Esta función mide la diferencia entre los resultados obtenidos en la clasificación de los datos y los resultados esperados [Agg18, LBH15].

2.3. Las Redes Neuronales

Llegado a este punto, se profundizará un poco más en dos modelos concretos del Deep Learning. Se van a presentar dos tipos de redes neuronales diferentes, las redes neuronales artificiales y las redes neuronales convolucionales y seguidamente serán comparadas entre ellas.

2.3.1. Redes Neuronales Artificiales

Las Redes Neuronales Artificiales, o Artificial Neural Networks (en adelante, *Artificial Neural Networks* (ANN)), son un modelo computacional del Deep Learning cuyo comportamiento trata de imitar el funcionamiento de una neurona biológica. Para su desarrollo, McCulloch y Pitts propusieron una unidad de umbral binario como modelo computacional para una neurona artificial [JMM96, Agg18].

Esta neurona matemática funciona computando la suma de sus N datos de entrada, todos ellos debidamente multiplicados antes por un peso correspondiente. Su resultado será 1 cuando esta suma sea superior a un umbral u o 0 si es inferior a dicho umbral.

McCulloh y Pitts probaron que, en principio, bien elegidos esos pesos permitían que una disposición síncrona de las neuronas realizase cálculos universales. La neurona de

McCulloh ha sido generalizada de muchas maneras. Una de las más usadas es el cambio de la función umbral por una función de activación, lo que mejora los resultados de dicha neurona.

Las ANN son vistas a veces como grafos ponderados dirigidos, dónde las neuronas artificiales son los nodos y las aristas ponderadas y dirigidas son las conexiones existentes entre cada neurona. Basándose en su arquitectura o patrón de conexión, pueden ser clasificadas en dos tipos [JMM96]:

1. **Red Neuronal Prealimentada.** También conocidas por su nombre en inglés, *Feed-Forward Networks*, se caracterizan por sus grafos, que no presentan bucles. Puede verse un esquema de este tipo de red en la Figura 2.3.

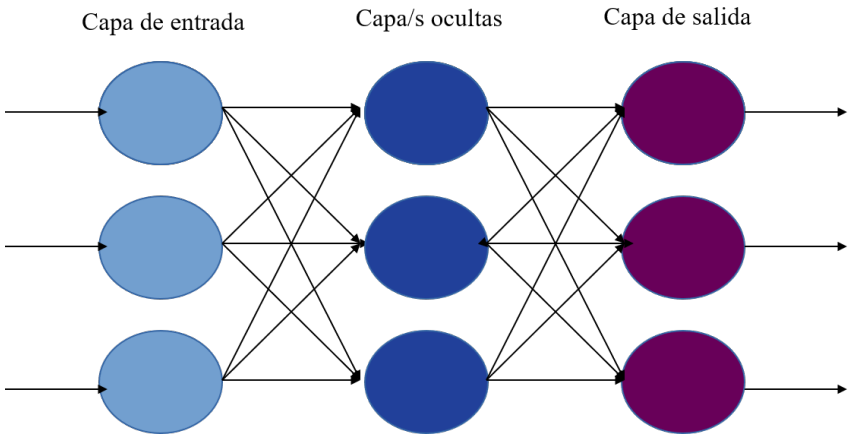

Figura 2.3: Esquema de una red neuronal prealimentada.

2. **Red Neuronal Recurrente.** Las conexiones de estas redes presentan bucles al ser posible que el resultado de una neurona sea el dato de entrada de una neurona anterior. Puede verse un esquema de este tipo de red en la Figura 2.4.

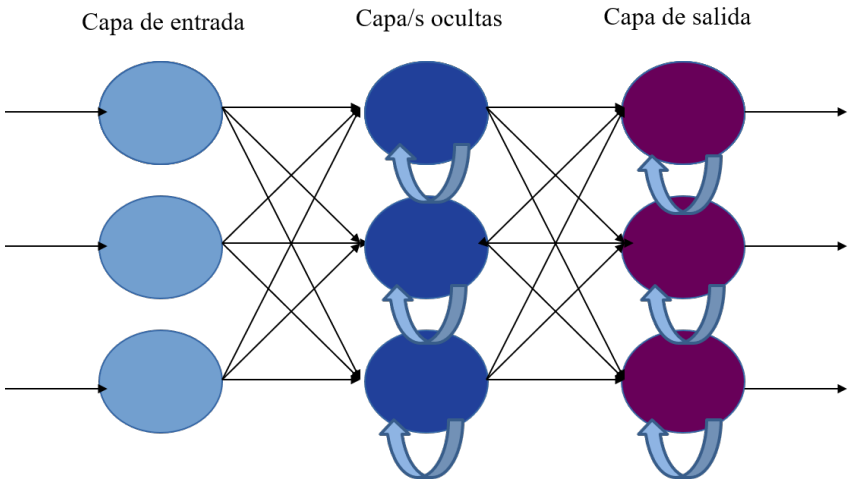

Figura 2.4: Esquema de una red neuronal recurrente.

Según sus conexiones, las redes neuronales pueden tener comportamientos muy diversos. Las redes *feed-forward* se caracterizan por ser estáticas, lo que significa que solo generan un conjunto de resultados para una entrada dada, en vez de una secuencia de estos. Además, estas redes no tienen memoria, ya que su resultado depende únicamente del estado de la neurona anterior. Las redes recurrentes son en cambio dinámicas, lo que implica que cada vez que aparece una nueva entrada, todos sus resultados deben calcularse de nuevo, ya que los datos de entrada pueden haberse modificado.

2.3.1.1. Aprendizaje con Redes Neuronales Artificiales

El proceso de aprendizaje en una ANN se lleva a cabo actualizando la arquitectura de la red y los pesos de cada conexión de forma que cada red pueda ser eficiente para una tarea específica. La red por lo general aprende los diferentes pesos de diferentes patrones de entrenamiento. De esta manera, su desempeño va mejorando según va probando distintos pesos y va actualizándolos según los resultados que ha obtenido en las pruebas anteriores. Esto es lo que concede a las ANN su mayor ventaja sobre el resto de sistemas expertos: su capacidad de aprender a partir de ejemplos, sin necesidad de un código complejo.

Sin embargo, el valor de los pesos no se descubre de forma mágica. Su manera de ir actualizando su valor, depende del algoritmo de aprendizaje que se haya elegido para el entrenamiento, que es el que determina que reglas han de seguirse a la hora de ajustar los pesos [JMM96, Agg18].

2.3.2. Redes Neuronales Convolucionales

Las Redes Neuronales Convolucionales, o más comúnmente conocidas con su nombre en inglés, Convolutional Neural Networks (en adelante CNN) son ANN profundas, diseñadas y pensadas especialmente para clasificar imágenes según lo que representan, agruparlas por similitud o reconocer objetos en una imagen. Las CNN realizan un reconocimiento óptico de imágenes para digitalizar texto y poder procesar así el lenguaje natural.

Las CNN tienen la capacidad de tomar una imagen y asignarle una importancia a cada una de sus características y/o aspectos, siendo capaz de diferenciar unos de otros. El tiempo de preprocesamiento que necesita es relativamente bajo, pues igual que los ANN, con suficiente entrenamiento, es capaz de aprender a diferenciar cada imagen, sin necesidad de un código extenso detrás [Agg18].

Su estructura es similar a las de una ANN, ya que, en el fondo, se trata de una *especialización* de una ANN. Sin embargo, hay más motivos por los que es más recomendable usar una CNN antes que una ANN. Si se compara una CNN con una red *feed-forward*, a priori, pueden parecer lo mismo. Si una imagen no deja de ser un conjunto de píxeles, ¿por qué no se transportan todos ellos a un array, y se usan como los datos de entrada para un perceptrón multicapa?

Aunque este método puede dar resultados óptimos con imágenes binarias extremadamente simples, en el momento en el que se intente reproducir con una imagen compleja, el método pierde por completo prácticamente toda su precisión. ¿Por qué? [TCC16]. La respuesta se encuentra en los propios píxeles. Si bien cada píxel es independiente de cualquier otro de forma absoluta, su posición exacta en la imagen puede proporcionar información extra debido a las dependencias con sus vecinos. Es decir, un píxel como tal puede dar la información de que tiene X ruido, pero eso no significa que el siguiente tenga el mismo ruido. Sin embargo, si un píxel tiene incorporada su posición en el espacio, puede dar la información extra de que, si tiene X ruido, es altamente probable

que sus vecinos más cercanos tengan un ruido muy similar, aún sin tener la información de dichos vecinos.

Y esta es la gran ventaja que presentan las CNN: este tipo de redes son capaces de capturar y entender las dependencias temporales y espaciales de los píxeles de una imagen, de forma que es capaz de aprovecharse de esa información extra para mejorar el desempeño del método. Esto lo hace a través de la aplicación de determinados filtros. Además, la arquitectura de las CNN facilita una mejor adaptación del conjunto de imágenes gracias a la reducción de parámetros y la reutilización de los pesos. De forma coloquial, se podría decir que una CNN es capaz de comprender el concepto de imagen de forma más inteligente.

Como resumen, el objetivo de una CNN es reducir el tamaño de las imágenes que se le pasan como datos de entrada para facilitar el proceso, pero sin perder por ello las características que son primordiales para una buena clasificación. Esto es importante, ya que no es solo buena aprendiendo características, sino también es escalable para bases de datos masivos [Agg18, RM15, TCC16]. La Figura 2.5 muestra un esquema de la arquitectura de este tipo de redes neuronales.

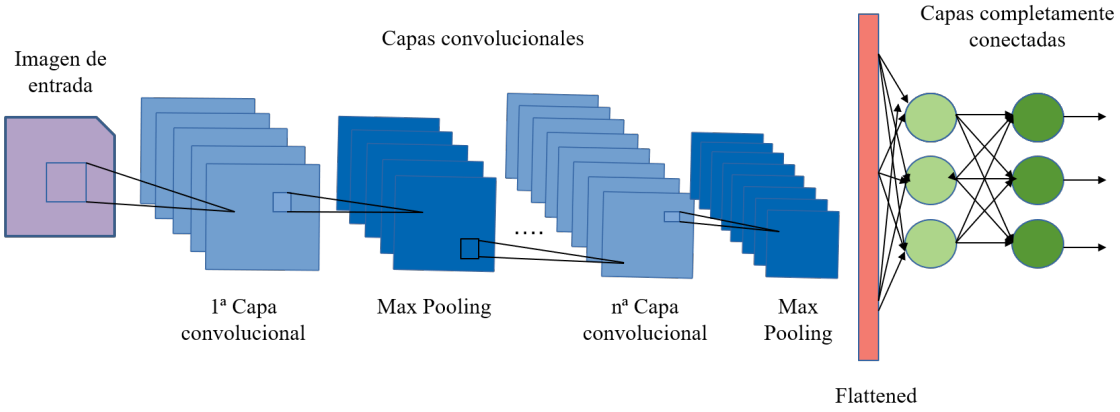

Figura 2.5: Esquema de la arquitectura de una red neuronal convolucional. En azul se muestran las capas convolucionales, en rojo, la capa *flattened* y en verde, las capas completamente conectadas.

2.3.2.1. Capas Convolucionales, Capas de Clasificación y el Kernel

Las CNN están compuestas por una o más capas, de las cuales la primera debe ser siempre una capa convolucional. La capa convolucional está compuesta por tres operaciones distintas: la convolución, la función de activación y el *pooling*. El resultado de esta capa es un mapa de caracteríticas de los datos introducidos.

La convolución puede expresarse de la siguiente forma [TCC16]:

$$a_j^l = \sum_{i=1}^{n} a_i^{l-1} * w_{ij}^{l-1} + b_j^l$$

donde * denota la convolución, a_j^l es la *j-ésima* salida de la capa l, $*w_{ij}^{l-1}$ es el kernel convolucional que conecta la *i-ésima* salida de la capa *l-1* con la *j-ésima* entrada de la capa l, b_j^l es el sesgo del parámetro de entrenamiento para la *j-ésima* salida de la capa l y n es el número de características de la capa *l-1*.

En este tipo de capas, el elemento que lleva a cabo la operación convolucional es llamado *kernel*. El kernel acostumbra a tener una profundidad igual a la de la imagen de entrada, especialmente con aquellas que tienen más de un canal.

La función de activación es la encargada de devolver una salida a partir de un valor de entrada. Se aplica a cada valor de la imagen filtrada y normalmente el conjunto de valores de salida se encuentra comprendido en un rango determinado como (0,1) o (-1,1). Lo ideal de una función de activación es que sean funciones en las que las derivadas sean simples, para minimizar con ello el coste computacional. Existen muchos tipos de funciones de activación [Agg18, RM15]:

- **Función sigmoide.** Transforma los valores introducidos en una escala (0,1). Sus ventajas son que permite interpretar cada resultado como un porcentaje de probabilidad en la clasificación y tiene un muy buen rendimiento en la última capa.

$$f(x) = \frac{1}{1 - e^{-x}}$$

- **Función Tangente Hiperbólica.** Se encarga de reescalar los valores introducidos a una escala (-1,1). Es similar a la sigmoide y es sus mejores resultados se obtienen con redes recurrentes.

$$f(x) = \frac{2}{1 + e^{-2x}} - 1$$

- **Función *Rectified Lineal Unit* (ReLU).** Esta función transforma los valores introducidos de tal forma que anula los valores negativos y deja los positivos tal y como entran. Esto provoca que no desactive demasiadas neuronas y proporciona un gran desempeño con imágenes y redes convolucionales.

$$f(x) = max(0, x)$$

- **Función Leaky ReLU.** Muy similar a la función ReLU, deja los valores positivos intactos y a los negativos los multiplica por un coeficiente de rectificación. Al igual que la ReLU, también presenta su mejor rendimiento con imágenes y redes convolucionales.

$$f(x) = \begin{cases} a * x & si \quad x < 0 \\ x & si \quad x \geq 0 \end{cases}$$

- **Función Softmax.** Esta función transforma las entradas de tal forma que sus salidas son una representación en forma de probabilidades, de tal manera que el sumatorio de todas las salidas de 1. Se utiliza para normalizar tipo multiclase.

$$f(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}$$

El siguiente paso importante es el *pooling*. Las capas de *pooling* suelen insertarse entre medias de dos capas convolucionales sucesivas. Su objetivo es reducir el tamaño espacial de la representación, para reducir así el número de parámetros con los que se opera y simplificar, pero sin perder su información. Normalmente, se realiza una media o un máximo.

El último proceso de la capa convolucional es la normalización de los mapas de características. Esto se hace para poder obtener valores comparables de cada neurona. Las capas de clasificación consisten en una o más capas completamente conectadas y una función *softmax* al final de éstas. Como puede deducirse de su nombre, en una capa completamente conectada, todas las neuronas están conectadas con todas las activaciones de la capa previa. Estas activaciones se pueden computar con una multiplicación matricial seguida de un sesgo de desplazamiento. Será la capa completamente conectada la que calcule el valor de clasificación que obtuvieron los datos con la ayuda de la función *softmax*.

2.3.2.2. Proceso de Aprendizaje

Cuando las características aprendidas pasan a través de una capa completamente conectada, son enviadas a la capa superior de la CNN, dónde se usará una función de activación *softmax* para clasificarlas. Para entrenar una CNN se usa un algoritmo de *backpropagation*. Éste hace que los sesgos y los pesos puedan ser modificados en las capas, tanto convolucionales como completamente conectadas, debido al proceso de propagación del error. De esta manera, el resultado de la clasificación puede retroalimentarse para extraer las características automáticamente y establecerse un mecanismo de aprendizaje.

La arquitectura de las CNN puede generar millones de parámetros debido a problemas de sobreajuste. Para evitarlo, se utiliza la técnica de *dropout*, también conocido como parámetro de aprendizaje. Consiste en establecer la salida de cada neurona oculta con una probabilidad de entre 0 y 0.5. Las neuronas que se eliminan de esta manera no transmiten su información a las capas posteriores y no participan en la propagación hacia atrás. Esta técnica aumenta la robustez, ya que una neurona no puede depender de la presencia de ninguna otra neurona. Por lo tanto, se fuerza a la red a aprender características más robustas que son útiles en conjunto [Agg18].

Capítulo 3

Análisis Forense de Identificación de la Fuente de Adquisición

Hoy en día, la posibilidad de adquirir un dispositivo que posea la capacidad de capturar imágenes se ha puesto al alcance de cualquiera. La masificación de estos dispositivos en la calle y la facilidad de poseer uno, llevan a la conclusión de que millones de imágenes son creadas todos los días en cualquier momento y en cualquier sitio.

Esto a su vez, genera un problema asociado: ¿cómo se puede saber de dónde procede una imagen en concreto? De hecho, ¿es posible identificar qué dispositivo fue el que capturó esa imagen? Tras muchos años de investigaciones y desarrollos, finalmente se puede concluir que sí, es posible identificar dicha fuente, gracias a la ayuda de diversos métodos del Machine Learning, entre ellos, las CNN.

Además, este incremento masivo de dispositivos, ha traído asociado un aumento del número de técnicas que permiten manipular una imagen y falsearla, con diversos métodos y resultados (eliminar objetos, añadirlos, fusionar dos imágenes, etc.). Claro está, este segundo problema implica que ya no es posible fiarse del contenido que muestra una imagen, y por tanto, no sería posible usarla como prueba irrefutable en un juicio, por ejemplo. Por ello, se plantea un segundo interrogante: ¿es posible detectar, de forma *rápida* e *infalible*, si una imagen ha sido trucada?

A día de hoy existen varios trabajos que desarrollan diversas técnicas para concluir si una imagen ha sido manipulada o no, valiéndose de la información de los píxeles de la propia imagen. Sin embargo, estas técnicas se encuentran en fase desarrollo todavía, y si bien hay varias propuestas ya y se ha demostrado las ventajas de unas sobre otras, este campo aún puede sorprender. Estos estudios se han desarrollado siguiendo dos caminos diferentes, cada uno de los cuales trata de dar respuesta a una pregunta diferente. En este capítulo se contestará a las preguntas formuladas y se explicarán las técnicas de detección de falsificación de imágenes y vídeos e identificación de la fuente de adquisición.

Este capítulo explicará ambos caminos, cada uno en un apartado diferente. Dentro de cada apartado, se presentará las diferentes formas en las que se pueden encontrar estas manipulaciones en los contenidos digitales, y a continuación, las técnicas que existen para identificarlos y detectarlos para cada uno de ellos.

3.1. Originalidad del Contenido que Presenta el Fichero Multimedia

Este primer planteamiento es la primera pregunta que se hace a la hora de afirmar si una imagen o vídeo es auténtico o no. Ser capaces de discernir si aquello que presenta una imagen o vídeo es real o ha sido manipulado puede ahorrar muchos quebraderos de cabeza y engaños. Pero, ¿cómo se puede asegurar que sí, que el contenido es original?

En primer lugar, será necesario conocer la existencia de los tres principales modos de alteración de contenido y el objetivo que persiguen cada uno de ellos.

3.1.1. Falsificación de Imágenes Digital

Las técnicas de falsificación digital de imágenes o Digital Image Tampering en inglés (*Digital Image Tampering* (DIT) en adelante) consisten en la manipulación del contenido original que presenta una imagen con la finalidad de esconder objetos o información que aparece en ella y engañar al observador para que se crea la verdad errónea de esa imagen. Para empezar, será importante conocer las diferentes formas en las que una imagen ha podido ser trampeada [BM13, MA13]:

- **Copy-Paste.** Esta técnica consiste en duplicar una sección concreta de la imagen y ubicarla en cualquier parte de la misma. Es de las técnicas más usadas y fáciles de reproducir. La Figura 3.1 presenta un ejemplo de esta técnica.

Figura 3.1: Ejemplo del efecto Copy Paste [RTD11].

- **Eliminar.** Es posible eliminar ciertos píxeles o partes de la imagen para borrar escenas u objetos. En la imagen siguiente, se puede ver como ejemplo como ha sido eliminado el ayudante de Mussolini para darle a la escena una actitud más heroica. La Figura 3.2 presenta un ejemplo de esta técnica.
- **Splicing.** Similar al *copy-paste*, esta técnica permite mover una zona concreta de la imagen para reubicarla en otro sitio diferente.
- **Retoques.** Comunes principalmente en las fotos de personas, consiste en alterar una zona de la imagen para alargarla, achatarla o similar, logrando así una perspectiva diferente a la real. También puede ser usada para alterar la textura de una superficie, como puede verse en este retoque fotográfico de Joan Crawford. La Figura 3.3 presenta un ejemplo de esta técnica.
- **Montajes.** Para realizarlo, será necesario la combinación de dos imágenes diferentes, normalmente, añadiendo elementos presentes en una de las imágenes en la otra. La Figura 3.4 presenta un ejemplo de esta técnica.

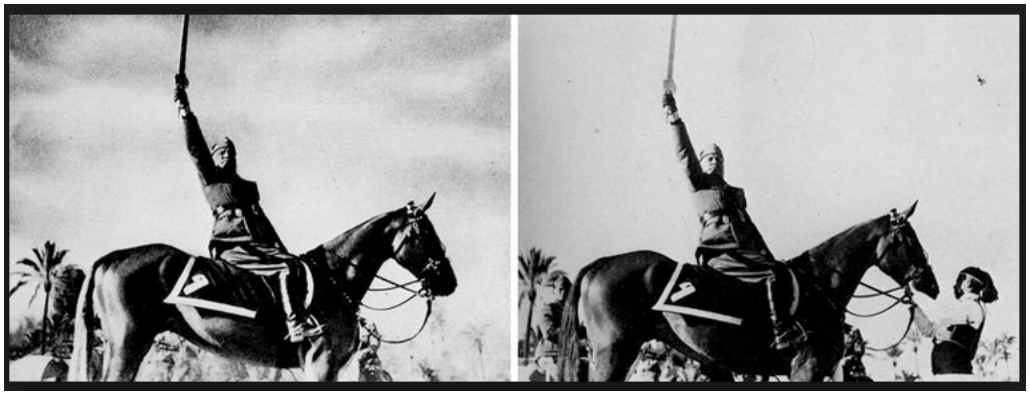

Figura 3.2: Ejemplo del efecto de eliminar un objeto de una imagen [MA13].

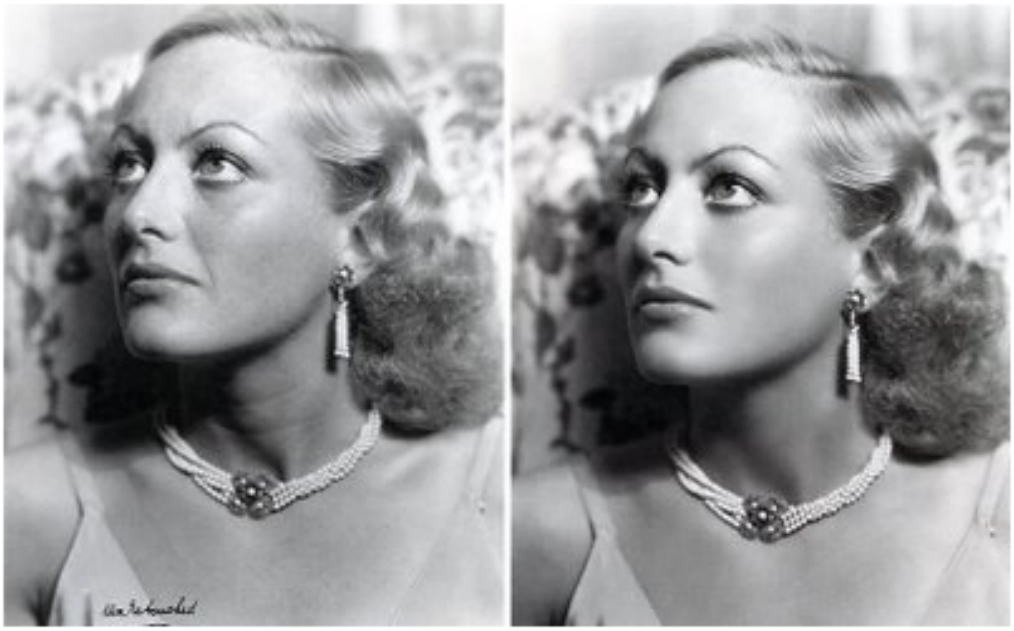

Figura 3.3: Ejemplo del efecto de retocar una imagen [RTD11].

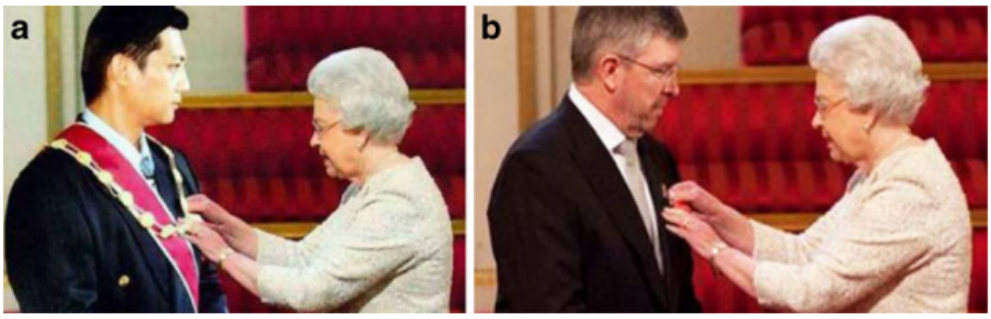

Figura 3.4: Ejemplo del efecto de realizar un montaje a partir de dos fotografías [BM13].

Descubrir si alguna de estas técnicas ha sido usada, es relativamente fácil si se tiene la inmensa fortuna de disponer la imagen original, ya que únicamente bastará con compararlas, incluso aunque la manipulación haya sido muy pequeña y sutil. Si no se tiene tanta suerte, será necesario recurrir a técnicas de detección de manipulación de imágenes. Las técnicas de detección para DIT se pueden dividir en términos generales en dos grupos: técnicas activas y técnicas pasivas o ciegas [MA13].

3.1.1.1. Técnicas Activas

Las técnicas activas de DIT requieren un preprocesamiento previo de la imagen y necesitan de la existencia de marcas de agua o firmas digitales en la imagen para identificarlas. Estas condiciones limitan su uso, especialmente la segunda, ya que para que dicha marca de agua exista, es necesario que la cámara que tomó la imagen añadiese dicha marca de agua en el momento de la captura, y que todas las cámaras existentes tuviesen una marca de agua propia.

Conseguir que todas las cámaras lleven incorporada dicha marca de agua, a día de hoy, no es viable por varios motivos. Poner de acuerdo a todas las compañías que fabrican cámaras de fotos para su incorporación es una ardua tarea que nadie quiere emprender por el momento. Además, sería necesario sacrificar calidad de cada imagen para añadir dicha marca, y es posible que en el proceso de añadir la marca cuándo se captura una imagen ralentice el proceso de captura de la misma. A todo ello, habría que sumarle que ya existen millones de cámaras en el mundo perfectamente operativas y sin su marca de agua correspondiente. Por lo tanto, esta idea es poco viable de poner en marcha en un futuro cercano.

Por todo ello, el uso de técnicas activas a la hora de detectar la falsificación digital de imágenes es muy poco común. Sin embargo, son el método más efectivo para detectar este tipo de falsificaciones y se han llevado a cabo numerosos estudios y experimentos en este área. Estas técnicas pueden clasificarse en dos categorías [MA13, RTD11]:

La primera utiliza una marca de agua *frágil*, que es capaz de detectar y localizar las modificaciones que se han realizado al contenido. Si bien este método tiene una tasa muy elevada de detección de manipulaciones, no es capaz de detectar de forma fiable un cambio en el brillo, el contraste o incluso si se ha añadido un objeto. Sin embargo, con un aumento de la escala de grises sí sería capaz de detectar una manipulación, aún cuando no es perceptible al ojo humano.

La segunda usa una marca de agua *semifrágil*, que solo detecta cambios significativos en la imagen y permite el procesamiento del contenido de la misma.

Para solventar los problemas y limitaciones de la marca de agua frágil, que no es capaz de distinguir una eliminación en una parte de la imagen de un simple ajuste del brillo, pero que en cambio es muy útil en las propiedades de localización y seguridad, apareció la marca de agua *híbrida*: una combinación de la marca de agua frágil y la robusta. Esta marca de agua híbrida es capaz de identificar con precisión los cambios así como de identificar las falsificaciones de otro tipo de operaciones. Actualmente, se sigue investigando en este campo con la finalidad de obtener un método fiable de identificación de falsificaciones que a su vez no perjudique a la calidad de la imagen tras su procesamiento.

3.1.1.2. Técnicas Pasivas o Ciegas

Las técnicas pasivas o ciegas son consideradas un desarrollo evolutivo en la ciencia de detección de falsificaciones en imágenes. A diferencia de las técnicas activas, estas no

requieren de información previa de la imagen ni de la presencia de ninguna marca de agua, lo que facilita su uso y que sean más conocidas.

Estas técnicas se basan en la asunción de que, tras haber sido alterada una imagen, es necesario que las propiedades estadísticas de la imagen hayan sido alteradas también inevitablemente. Por tanto, para su estudio no es necesario tener la imagen original ni una marca de agua presente en la imagen, basta con estudiar estas propiedades estadísticas de la misma para obtener una conclusión. Principalmente, se suelen enfocar en dos métodos pasivos [MA13, RTD11, SM08]:

- **Detección de clonación**: El arte de copiar y pegar una parte de la imagen en otro sitio para ocultar algo es una de las técnicas más empleadas en la falsificación. Tanto, que muy bien llevada a cabo es prácticamente imposible de detectar por el ojo humano. Tampoco es posible realizar una búsqueda computacional, ya que como la región clonada puede ser de cualquier forma y tamaño, y ubicarse en cualquier punto de la imagen, vuelve imposible una búsqueda exhaustiva en todas sus posiciones. Sin embargo, se ha encontrado que este tipo de falsificación introduce una correlación entre el segmento de la imagen original y el pegado. Estos segmentos sólo pueden coincidir de forma aproximada, pero nunca exacta, debido al uso de operaciones de posprocesado como el suavizado, que son imprescindibles para una buena clonación. También pueden haber sido afectados por una comprensión. Esto es lo que se usa como base para la detección de este tipo de trampa y asegura su éxito.

- **Detección de técnicas de unión (*splicing*)**: La unión de dos o más imágenes en una sola es otra de las técnicas de falsificación más populares. Igualmente, si es realizada con meticulosidad, los bordes de las dos imágenes unidas no pueden ser detectados a simple vista. Es utilizada principalmente para distorsionar el contenido semántico de la imagen y crear *una segunda realidad*. En cambio, esta técnica no precisa de ninguna técnica de posprocesado como suavizado o desenfoque. Al no depender de esto, y sumado al hecho de que la imagen no presenta regiones duplicadas, se llega a la conclusión de que esta técnica es mucho más compleja de detectar que una simple clonación. Sin tener las imágenes originales de la unión como referencia, no se dispone de ninguna pista. Sin embargo, afortunadamente se ha encontrado que estas uniones alteran las estadísticas de Fourier de orden superior, proporcionando una ayuda para detectar estas trampas. Actualmente se están llevando a cabo numerosas investigaciones y descubrimientos sobre cuál es el método más eficaz para encarar este problema.

3.1.2. Esteganografía Digital de Imágenes

Esta otra técnica manipulativa, si bien altera el contenido de la imagen, no busca engañar al espectador con dicho contenido. Su propósito va más allá: la finalidad de esta técnica consiste en añadir una información en la imagen, que estará escondida a simple vista. En otras palabras, la esteganografía de una imagen se puede definir como el proceso de añadir una información secreta a una imagen. Esta información deberá ser codificada dentro de los bits de dicha imagen, de forma que a simple vista no se aprecie este añadido y no levante sospechas [MA13]. En la Figura 3.5 puede verse un ejemplo de esteganografía. Debido a su objetivo, es necesario que la imagen original y la imagen *stego* sean lo más similar posible, al contrario que las imágenes obtenidas por falsificación digital, cuyo propósito es obtener una imagen diferente de la original.

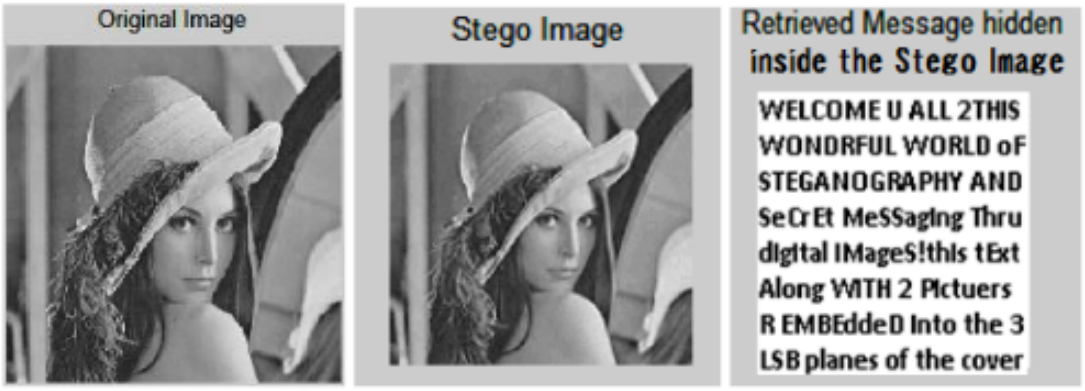

Figura 3.5: Ejemplo de una imagen con esteganografía [MA13]

3.1.3. Falsificación de Vídeos

Las falsificaciones en los vídeos, se pueden dar de diferentes maneras:

- Alterando, eliminando o cambiando su audio por el de cualquier otro vídeo.
- Modificando el contenido de uno o más fotogramas. En este caso, las técnicas usadas para la falsificación son prácticamente iguales que las empleadas en fotografía que hemos visto anteriormente.
- Eliminando, añadiendo o reordenando uno o más fotogramas del vídeo.

Al igual que se han propuesto numerosas técnicas para detectar falsificaciones en imágenes, también se han propuesto algunas para detectar esas mismas falsificaciones en vídeos. Estas técnicas se han dividido en dos estrategias [KAS17]:

- **Estrategia Activa**: Las estrategias forenses activas se caracterizan por determinar la autenticidad del contenido del vídeo con la ayuda de la presencia de una marca de agua en el propio vídeo, como en el caso de las imágenes. Por desgracia, estas técnicas tienen los mismos problemas que las imágenes, por lo que su utilización es igual de complicada de extender, y poco populares por ello.
- **Estrategia Pasiva**: Las estrategias forenses pasivas analizan características específicas, bien estáticas, bien temporales, que surgen debido a la intromisión de operaciones de falsificación presentes en el vídeo. Este enfoque es utilizado para detectar cualquier manipulación no autorizada que se ha llevado a cabo en los siguientes puntos [KAS17]:
  - **En un fotograma.** Las falsificaciones que se llevan a cabo en el *interior* de un fotograma son aquellas que manipulan un único fotograma, ya sea uno o varios píxeles de él, objetos o el fotograma entero. Pueden llevarse a cabo con eliminaciones, *copy-paste* y/o montajes.
  - **Entre dos o más fotogramas.** Este tipo de falsificaciones son muy fáciles de llevar a cabo, pero prácticamente imposibles de detectar sin la ayuda de alguna técnica especializada. Para crearlas, basta con insertar y/o eliminar uno o varios fotogramas completos del vídeo en puntos concretos, sin necesidad

de tocar su contenido. Para la identificación de este tipo de falsificaciones se han propuesto varias técnicas. Casi todas las técnicas de falsificación tienen en común unos mismos pasos: en primer lugar, es habitual extraer del vídeo una serie o secuencia de fotogramas del mismo, a las que se les aplica alguna técnica como eliminación, *copy-paste* o montaje y finalmente, se reconstruye el vídeo. Esta reconstrucción deja tras de sí una comprensión, que sumada a la original, se convierte en una comprensión doble. Por este motivo, muchas de las técnicas de detección se basan en análisis de la compresión MPEG que presenta el vídeo para determinar si puede haber sido manipulado o no. Sin embargo, más adelante se determinó que esta compresión puede presentarse también en vídeos que no han sido manipulados, pero que han sido transmitidos a través de alguna red social o descargado de algún sitio. Por ello, se vio la necesidad de encontrar la presencia de algún otro elemento que delatase la alteración de un vídeo, similares a las que se utilizan con las fotografías, como la variación del brillo en los *frames* o el análisis de la perturbación en la correlación temporal entre fotogramas.

3.2. Identificación de la Fuente de Adquisición

El segundo camino de investigaciones trata de resolver esta pregunta para determinar si una imagen es auténtica o no. Sin embargo, ¿en qué afecta a una imagen manipulada qué cámara la capturó?

Identificar la fuente de adquisición de una imagen puede dar más información sobre la misma de la que a priori se puede pensar. Por ejemplo, se pone el caso de analizar una imagen de dos maneras: intentando descubrir una falsificación en ella y descubriendo qué dispositivo la capturó. El primer análisis da como resultado que la imagen no ha sido manipulada. En cambio, el segundo revela que se han encontrado dos cámaras diferentes como fuente de adquisición. Lógicamente, eso, en una imagen no manipulada, es imposible. Luego, si los dos análisis son correctos, solo queda una deducción posible: el averiguar la fuente de adquisición de una imagen da una información extra acerca de las falsificaciones que en algunas ocasiones, otras técnicas no son capaces de detectar. Además de esta utilidad, este tipo de identificación puede resultar útil en otros aspectos:

Determinar qué dispositivo realizó una determinada fotografía puede ayudar también a sospechar si una imagen es lo que dice ser. Por ejemplo, si se quiere usar una imagen en un juicio como prueba: si se presenta una imagen en la que se ve una localización fácilmente reconocible (las torres Kío de Madrid, por ejemplo), pero se demuestra que el dispositivo que supuestamente hizo la fotografía no estuvo allí en ese momento, bien porque otra foto lo demuestra o por otro motivo, ya existen razones suficientes para sospechar (y poder demostrar después) que esa foto es posiblemente una falsificación.

Otro uso interesante en la misma línea es la identificación de la fuente para poder encontrar su procedencia y así al autor de un delito. Este uso se da especialmente en delitos en los que el delincuente graba sus acciones y las cuelga en la red o las distribuye por las redes sociales. Identificando el modelo de dispositivo que lo grabó, las autoridades pueden restringir de forma considerable el número de sospechosos y encontrar de manera más fácil y rápida al verdadero autor del delito.

3.2.1. Técnicas de Identificación de la Fuente de Adquisición en Imágenes

A día de hoy, se han propuesto múltiples formas de identificar la fuente de adquisición de una imagen. Todas ellas se basan en el estudio de una o más características de los componentes físicos que integran la cámara y que bien hayan podido dejar algún tipo de rastro en la imagen que pueda ser estudiado o bien hayan dejado su huella en los propios metadatos (o *pipeline* en inglés) de la imagen al crearla. De esta manera, tratan de recopilar las características específicas de forma manual o automática y distinguir así cada modelo o dispositivo. [KG15, YNZZ19, GZY+18, MAU15]. Para lograr una buena identificación de las cámaras, se han estudiado principalmente los siguientes elementos:

3.2.1.1. Sensor de la Cámara

El elemento fundamental de una cámara para capturar una imagen es su sensor. Él es el encargado de detectar y capturar la información que compondrá la imagen y es por tanto el elemento que más información y rasgos propios aporta a la fotografía. Existen varios métodos basados en su estudio para identificar la fuente de adquisición de una imagen [FFG08].

- **Ruido de respuesta no uniforme.** Mucho más conocido por su nombre inglés, el *Photo-Response Non Uniformity* (*Photo-Response Non-Uniformity* (PRNU) en adelante) es uno de los métodos de estudio del sensor que proporciona parte de los mejores resultados.

  El PRNU es definido como el componente principal de la huella de una cámara, siendo una característica específica en cada cámara debido a las imperfecciones en el proceso de manufacturación. Gracias a él, es posible detectar un patrón multiplicativo de ruido en cada imagen obtenida con una cámara, y es especialmente interesante ya que es específico de cada dispositivo en vez de cada modelo.

- **Ruido de patrón fijo.** Este tipo de ruido, o *Fixed Pattern Noise*, (*Fixed Pattern Noise* (FPN) en adelante) es un patrón de ruido particular presente en los sensores de las imágenes digitales. Suele ser apreciable en imágenes cuyo disparo ha sido de larga exposición dónde píxeles particulares son susceptibles de tener intensidades superiores al ruido general del fondo.

- **Polvo en el sensor.** La cavidad que guarda al sensor de una cámara no es hermético. Por lo tanto, con el paso del tiempo, el sensor es proclive a ensuciarse, acumulando polvo y otros pequeños elementos. Una de las propuestas más modernas es el estudio del patrón del polvo acumulado en el sensor de las cámaras réflex de lente única (DSLR).

3.2.1.2. Lentes

El otro gran elemento imprescindible a la hora de tomar una fotografía es la lente empleada para ello. Cada lente posee unas características propias como puede ser la distancia focal o la apertura, que condiciona la fotografía tomada y a su vez imprime en ella sus características como si fuese una huella digital. Basados en las lentes, se han estudiado las siguientes características:

- **Distorsión radial.** Esta característica intrínseca de la lente puede ser utilizada como identificador, ya que cada modelo presenta un único patrón de distorsión [CLW06].

3.2.2. Técnicas de Identificación de la Fuente de Adquisición en Vídeos

Según [BFM+12], las técnicas de análisis forense de los vídeos se pueden dividir en tres grupos, según cuál sea su objetivo: identificar la fuente de adquisición, determinar si un vídeo ha sido comprimido o no e identificar si ha sido falsificado o alterado.

Este trabajo se centrará en el primer objetivo: la identificación de la fuente de adquisición de un vídeo. Al igual que con las imágenes, el análisis de la fuente de adquisición de un vídeo tiene como objetivo identificar el origen del que procede un vídeo. Sus técnicas han sido menos estudiadas y desarrolladas que las que se usan con las fotografías. Sin embargo, muchas de esas mismas técnicas pueden ser aplicadas igualmente a los fotogramas de un vídeo. [SAR+13] divide estas técnicas en cinco grupos:

- **Técnicas Basadas en Metadatos**: Estas técnicas, si bien son las más sencillas de analizar, dependen de la presencia de metadatos en los fotogramas. Esto no siempre se da, ya que que el fabricante añada metadatos no es obligatorio.
- **Técnicas Basadas en Características de la Imagen**: Las características de la imagen que se usan para tratar de identificar la fuente de adquisición son principalmente, las tres siguientes:
  - Características de color, ya que las tonalidades de color que se agregan a las imágenes pueden variar ligeramente de un fabricante a otro.
  - Métricas de calidad de la imagen, o más conocidas por su nombre en inglés: *Image Quality Metrics* o *Image Quality Metric* (IQM).
  - Estadísticas del dominio de la frecuencia o *wavelet*.

  Para analizar este tipo de características, con frecuencia se emplea una *Support-Vector Machine* (SVM).
- **Técnicas Basadas en Defectos de la Matriz CFA e Interpolación Cromática**: La matriz *Color Filter Array* (CFA) toma su nombre de sus siglas en inglés, Color Filter Array. Esta matriz de filtros de color es un mosaico de minúsculos filtros de color colocados sobre los píxeles de los sensores de imagen para capturar la información del color. Las diferencias marcadas que presentan estas matrices y la especificación de los algoritmos de interpolación cromática en los diferentes modelos de cámaras permiten su uso para la identificación de las mismas.
- **Técnicas Basadas en Imperfecciones del Sensor**: Las técnicas basadas en las imperfecciones del sensor se han ido separando poco a poco en dos familias bien diferenciadas: aquella basada en los defectos del píxel y la que se encarga de analizar el patrón de ruido en el sensor. Esta es la más interesente. Se basa en las demostraciones de [LFG06], que demostró que las cámaras generan un patrón de ruido o *Sensor Pattern Noise* (*Sensor Pattern Noise* (SPN)) que puede ser utilizado como método único de clasificación. Posteriormente, [Li10] probó que el ruido extraído del sensor podía ser contaminado por los detalles de las imágenes, por lo que propuso una mejora del método para atenuar esta influencia y aumentar la precisión de la identificación.
- **Técnicas Basadas en las Transformadas *Wavelet***: En estas técnicas existen varios enfoques propuestos por diferentes autores. Destacan la identificación basada en las características de probabilidad condicional, que se propuso con la intención

de detectar mensajes ocultos en las imágenes, y el uso del patrón de ruido del sensor conjuntamente con la transformada *wavelet*. Existen otros métodos para el análisis de la fuente de adquisición de un vídeo que se basan en la secuencia de codificación del mismo, y que se pueden aplicar sin necesidad de extraer los fotogramas. Un ejemplo de ellos es el trabajo de [SXD09], que propone un algoritmo basado en la información del vector de movimiento en el flujo codificado.

3.2.3. Identificación de la Fuente de Adquisición con Redes Neuronales Convolucionales

Se ha demostrado que es posible llevar a cabo la identificación de la fuente de adquisición de una imagen ayudándose del uso de CNN. Para ello, se necesita la extracción de un vector de características de los datos. Aquí, el modelo es construido por un algoritmo de clasificación que conoce las características y lo construye en base a ellas. Para la identificación de la cámara, el clasificador evalúa la proximidad del modelo previamente aprendido y la imagen.

Sin embargo, a pesar de todas las técnicas punteras que se utilizan en estos estudios, el enfoque de las CNN para la identificación del modelo de cámara no es muy usado aún. El enfoque general del Machine Learning es obtener una buena representación de los datos de entrada y poder generalizar unos patrones de aprendizaje. Ya que una buena representación de los datos puede derivar en una ejecución con muy buenos resultados, la clave está en conseguir esa buena representación, lo cual puede ser bastante costoso en tiempo y esfuerzo, y puede derivar incluso en un campo totalmente diferente y específico.

¿Por qué entonces es interesante el uso de las CNN para identificar el modelo de una cámara? Los algoritmos del Deep Learning son una herramienta prometedora a la hora de automatizar la representación de datos complejos a un alto nivel de abstracción. El Deep Learning ofrece una gran ventaja a la hora de analizar datos no supervisados con muy buenos resultados, pero tiene como inconveniente que requiere de sofisticados recursos informáticos como una GPU y una ingente base de datos.

Utilizando una CNN como una *caja negra*, a la cual se proporcionan unos datos y devuelve unos resultados sin saber qué procedimiento ha llevado a cabo en su interior, se obtiene un rendimiento muy débil para la identificación de la fuente de adquisición de un contenido multimedia. Para lograr una mejora en su desempeño y unos resultados aceptables, habrá que indicarle a esa CNN qué características son necesarias e imprescindibles que tenga en cuenta en su clasificación [ZZC+17].

Capítulo 4

Estado del Arte

En este capítulo se hará un resumen rápido de todas las investigaciones más relevantes que se han llevado a cabo a día de hoy en el estudio de la identificación de la fuente de adquisición de una imagen o un vídeo. Los enfoques que se le han dado a las investigaciones para identificar la cámara fuente pueden clasificarse en dos grupos. Para facilitar la explicación, se hablará de cada grupo en un subapartado distinto.

4.1. Tipos de Enfoque en las Investigaciones

Las investigaciones acerca de los métodos usados para identificar la fuente de adquisición de una imagen o vídeo han sido divididas de forma consensuada en dos grupos:

4.1.1. Basado en Modelos

El primer grupo recoge todos los métodos que requieren computar un modelo para identificar a la cámara que creó la imagen que se esté analizando. Esto quiere decir que para ser capaces de llegar a una conclusión satisfactoria, es necesario buscar una traza o característica específica en la imagen analizada acorde con un modelo hipotético que se conozca a priori. Una vez obtenido dicho modelo, se evalúa la proximidad estadística que existe, como por ejemplo, la correlación que existe entre la imagen y el modelo desarrollado. Con esta técnica, se han llevado a cabo varios avances importantes:

- **Lentes**: Uno de los modelos más aceptados y usados es el estudio de las lentes y sus características. De esta manera, es posible aprovecharse de sus propiedades en la tarea de identificar una cámara en cuestión. [CLW06] han demostrado que es posible obtener una precisión muy alta en la identificación a partir de medir la distorsión radial intrínseca de las lentes de cada cámara. Su estudio se basa en el hecho de que la gran mayoría de las cámaras poseen lentes. Éstas tienen una superficie bastante esférica y su distorsión radial sirve como huella única para su identificación. Para terminar, entrenan y testean una máquina de soporte vectorial como clasificador utilizando parámetros obtenidos de las medidas de aberración.

- **Sensores**: [CR10] han definido la lectura del ruido como una importante característica intrínseca del sensor de una cámara digital. Un detalle importante acerca de este ruido es que no puede ser borrado. Por ello, han propuesto un estudio que mide el ruido del sensor de una imagen, usando el promedio de la desviación para resolver este problema. [ZZC+17] realizan un novedoso estudio investigando la creación de una CNN prealimentada que sea capaz de eliminar el ruido de

una imagen, aprovechando todos los avances existentes en arquitectura profunda y algoritmos de aprendizaje. Este trabajo propone un modelo de CNN capaz de manejar la eliminación de ruido gaussiano en una imagen con un nivel de ruido desconocido.

4.1.2. Basado en Datos

El segundo grupo se caracteriza por el uso de métodos que se basan en la extracción de un vector de características y en los algoritmos de Machine Learning tradicionales. Esto quiere decir que el objetivo es aprender las características que poseen las imágenes disparadas con diferentes cámaras directamente desde las propias imágenes, sin imponer un modelo preconcebido. Al igual que en el otro grupo, se han llevado a cabo diversos trabajos y estudios enfocados de formas distintas para resolver el mismo problema: encontrar la fuente de adquisición de una imagen.

- **Aprendizaje Supervisado**: [KSM04] han desarrollado un método basado en el aprendizaje supervisado que se vale de las características extraídas de los dominios espaciales y ondículas. Para la clasificación de cinco cámaras distintas, utilizan una máquina de soporte vectorial, obteniendo una tasa de error comprendida entre el 5 % y el 22 %.

- **Sensores**: Al igual que en el otro grupo, los sensores y sus características pueden ser empleados en las investigaciones. El PRNU, como se ha descrito en el capítulo anterior, es uno de los métodos de estudio del sensor que proporciona parte de los mejores resultados. [FFG08] se basaron en el concepto de que el PRNU es único en cada cámara con el objetivo de demostrar que gracias a él, se podía identificar no sólo la marca de un dispositivo, sino también el propio dispositivo. Esto es posible ya que las imágenes en formato TIFF/JPEG contienen una estructura local específica debido a varios procesos producidos durante la toma de la fotografía. Esta estructura puede ser detectada extrayendo una serie de características numéricas y clasificarla utilizando métodos de clasificación de patrones. Su clasificación usando estas características arroja una tasa de error del 11.2 % a la hora de identificar las cámaras. Otro ejemplo son los trabajos de [LWY15], que han realizado una extensa evaluación de las diferentes maneras de mejorar el patrón de ruido en el sensor (*Sensor Pattern Noise* o SPN). Gracias a esto, han identificado qué métodos ofrecen alguna mejora y proporcionan algunas ideas sobre el comportamiento de los SPN. Consiguieron una tasa de error del 15 %. En esta misma línea de trabajo, [LFG06] han conseguido resultados muy interesantes en el coste de una gran demanda computacional, a la vez que [CPN+17] han propuesto una implementación escalable y distribuida factible de estos resultados.

- **Redes Neuronales Convolucionales**: Si bien todos los trabajos que han utilizado esta técnica pertenecen al enfoque del Machine Learning, al ser este estudio sobre CNN, dedicaremos un subapartado entero a estos trabajos.

4.2. Uso de las Redes Neuronales Convolucionales

Como se ha visto a lo largo de todo este estudio, el Deep Learning y en particular las CNN, han demostrado un muy buen desempeño en múltiples tareas como reconocimiento facial, detección de peatones o reconocimiento de escritura. Al contrario que todos los

trabajos anteriores basados en la extracción de vectores de características y en los algoritmos clásicos de Machine Learning, el Deep Learning depende de la capacidad de los datos de entrada de impulsar su propio procesamiento de características. Gracias a las indiscutibles ventajas de las CNN, como minimizar la función de coste durante el proceso de entrenamiento, las CNN son capaces de detectar patrones en los datos de entrada. Este problema ha sido ya estudiado por varios grupos de investigadores usando el Deep Learning:

4.2.1. Identificación de los Modelos de Cámara Tradicionales

El primer reto que se planteó la comunidad científica fue el de identificar la fuente de adquisición de una imagen tomando como muestras las imágenes capturadas por cámaras fotográficas convencionales. Aquí se puede encontrar cámaras compactas, réflex, cámaras sin espejo o *bridges*, entre otras. Los trabajos más relevantes se presentan a continuación.

4.2.1.1. Trabajos y Estudios

[BBBT16] proponen una red convolucional con dos objetivos:

1. **Reconocimiento de dispositivo.** Es decir, pretenden que la red convolucional sea capaz de identificar imágenes tomadas por diferentes dispositivos pertenecientes a un mismo modelo.

2. **Reconocimiento de modelo.** De forma similar al objetivo anterior, aquí los autores tratan de identificar fotografías tomadas por el mismo modelo de cámara, sin tener en cuenta qué dispositivo fue el que la tomó.

Para el entrenamiento y testeo de la red se han valido del *dataset Dresden* [GB11], compuesto por 16k imágenes, tomadas por 74 dispositivos distintos pertenecientes a 27 modelos de cámara diferentes. Este *dataset* ha sido usado íntegro en los experimentos de cada objetivo, de forma separada, por lo que es necesario realizar una copia del *dataset*. El 70 % de las imágenes serán usadas para el entrenamiento y el 30 % restante para la validación de la red. Además, cada imagen es recortada en 25 regiones o *patches* de 48x48 píxeles.

[TCC16] llevan a cabo una evaluación de la mejora obtenida al modificar la red que propone [KSH12]. También comparan experimentalmente su red con AlexNet y GoogleNet. [TCC16] utilizan para sus experimentos dos *dataset* diferentes, compuestos por 33 modelos de cámaras: el primero es el *dataset Dresden*, el mismo que utilizan [BBBT16]; y el segundo es un *dataset* propio de ellos, compuesto por 6 modelos de cámaras personales. Antes de manipular las imágenes, estas son divididas en un 80 % para entrenamiento y las restantes para validación. A continuación, son recortadas en un tamaño de 256x256 píxeles, lo que aumenta el tamaño del *dataset* con el que operan.

4.2.1.2. Resultados y Propuestas

[BBBT16] proponen como resultado final a sus investigaciones una red convolucional formada por tres capas convolucionales y dos capas completamente conectadas, como se aprecia en la Tabla 4.1.

#	Tipo de capa	Parámetros	Dimensiones
1	Convolucional	Tamaño Kernel	7x7x3
		# de filtros	128
		Stride	1
2	ReLU	-	-
3	Max Pooling	Tamaño Kernel	3x3
		Stride	2
4	Convolucional	Tamaño Kernel	7x7x128
		# de filtros	512
		Stride	1
5	ReLU	-	-
6	Max Pooling	Tamaño Kernel	3x3
		Stride	2
7	Convolucional	Tamaño Kernel	6x6x512
		# de filtros	2048
		Índice de dropout	0.5
8	ReLU	-	-
9	Completamente Conectada	Tamaño Kernel	1x1x2048
		# de filtros	2048
		Stride	1
10	ReLU	-	-
11	Completamente Conectada	Tamaño Kernel	1x1x3048
		# Combinaciones	# Clasificaciones
12	SoftMax	-	-
10	LogLoss	-	-

Tabla 4.1: Detalle de la red propuesta por [BBBT16]

Esta estructura les permite conseguir una precisión del 94 % en sus experimentos para identificar el modelo. Sin embargo, como ellos mismos reconocen, no es suficientemente complejo, por lo que la precisión que obtienen a la hora de identificar los dispositivos es únicamente de un 29.8 %.

[TCC16] realizan varios experimentos con su red una vez definida la arquitectura definitiva de la red convolucional. Esta arquitectura está formada por tres capas convolucionales y una capa *single pooling* a continuación de éstas. Los detalles de esta red pueden verse en la Tabla 4.2.

Para comparar su estructura con AlexNet y con GoogleNet, [TCC16] entrenan esas dos redes con el mismo *dataset* que han usado ellos con su red. Una vez realizados los experimentos, la tasa de error que obtienen es menor del 10 %.

4.2.2. Identificación de Cámaras de Móviles

Tras los estudios realizados en la identificación del modelo de la cámara fotográfica que tomó una determinada imagen, algunos estudios van más allá y repiten estos experimentos con las imágenes tomadas por un móvil o una tablet en algunos casos. El objetivo de estos estudios es ser capaces de identificar el modelo de un móvil que tomó una imagen o incluso si fue su cámara delantera o la trasera.

4.2.2.1. Trabajos y Estudios

[FONBCS18] proponen desarrollar una arquitectura de CNN que sea capaz de clasificar de forma satisfactoria qué modelo de móvil realizó una determinada fotografía. Para sus

#	Tipo de capa	Parámetros	Dimensiones
1	Convolucional	Tamaño Kernel	3x3
		# de filtros	256
		Stride	2
2	ReLU	-	-
3	Convolucional	Tamaño Kernel	3x3
		# de filtros	63
		Stride	2
4	ReLU	-	-
5	Convolucional	Tamaño Kernel	3x3
		# de filtros	63
		Índice de dropout	-
6	ReLU	-	-
7	Max Pooling	Tamaño Kernel	3x3
		Stride	2
8	Completamente Conectada	Tamaño Kernel	-
		# de filtros	256
		Stride	1
9	Completamente Conectada	Tamaño Kernel	-
		# de filtros	4096
		Stride	1
10	Completamente Conectada	Tamaño Kernel	-
		# Combinaciones	# Clasificaciones
11	SoftMax	-	-

Tabla 4.2: Detalle de la red propuesta por [TCC16]

experimentos han utilizado el *dataset MICHE-I*, que consiste en 3732 imágenes de tres dispositivos distintos: un Iphone 5, un Samsung Galaxy IV y un Galaxy Tablet III. En ellos, todas las imágenes son divididas en 256 *patches* de 32x32 píxeles. Este trabajo tiene dos objetivos principales:

1. **Reconocimiento del modelo.** Se trata de identificar cuál es el modelo de móvil que realizó una determinada fotografía.

2. **Reconocimiento de la cámara.** Esta parte del trabajo intenta identificar cuál de las cinco cámaras (cada móvil posee dos, una delantera y otra trasera, y la tablet posee solo una) realizó la foto analizada.

[ABHAN+19] crea su propio *dataset* de 3900 imágenes, usando 3 modelos de cámara diferentes y utilizan la transferencia de aprendizaje para extraer las características y acelerar así el proceso. También añaden otros modelos del *dataset Kraggle* para aumentar el tamaño del suyo propio. En su trabajo, estudian tres clasificadores diferentes de Machine Learning para descubrir cuál funciona mejor en este problema. Los clasificadores usados son:

- Máquina de Soporte Vectorial.

- Regresión Logística.

- Bosques aleatorios.

4.2.2.2. Resultados y Propuestas

Tras llevar a cabo sus investigaciones, [FONBCS18] concluyen que la mejor estructura para la red está formada por dos capas convolucionales en vez de tres, y que el kernel de estas es recomendable que sea pequeño. También utiliza una capa *max pooling* y tres completamente conectadas. Con esta arquitectura es capaz de obtener un 98.1 % de precisión en la identificación del modelo, pero solo un 91.1 % en la precisión de identificación del sensor, lo que lleva a pensar que es mucho más difícil distinguir cámaras del mismo fabricante. A su vez, viendo la matriz de confusión, podemos ver que tampoco es fácil distinguir entre cámaras situadas en la misma posición.

Para finalizar, [FONBCS18] compara sus resultados con los de [BBBT16] y los de [TCC16]. Prueba ambas estructuras con el *dataset MICHE-I*, puesto que estas arquitecturas no habían sido probadas con imágenes tomadas por móviles. Tras los experimentos, puede ver que su estructura es la más eficiente de las tres.

Con los resultados obtenidos por [ABHAN$^+$19], se llegó a la conclusión de que el mejor clasificador en este problema fue la SVM, llegando a mostrar una precisión del 98.82 %, seguida por la Regresión Logística, que obtuvo una precisión del 98.54 %. La peor parada en estos experimentos fue el Bosque Aleatorio, que sólo mostró un 97.16 % de precisión en sus clasificaciones. Sin embargo, a pesar de los buenos resultados obtenidos con la SVM, [ABHAN$^+$19] señalan que este método de clasificación fue sin duda el más lento, con una tardanza de poco menos de 30min frente a los 6 o 7min que tardaron los otros dos métodos. También remarcan la pérdida de precisión en los modelos de dispositivos que presentaban menos ejemplos a la hora de entrenar.

4.2.3. Identificación de la Fuente de Adquisición de Vídeos

Finalmente, se hablará de aquellos trabajos más relevantes y punteros en la identificación de la fuente de adquisición de un vídeo. Los vídeos, hablando de forma muy general, son una continua sucesión de imágenes llamadas fotogramas, que son las encargadas de generar la sensación de movimiento que se ven en ellos. Es por ello que a lo largo de todo este trabajo se ha hablado de imágenes, pues no es posible comprender los vídeos y su identificación sin comprender el nivel inferior que son las imágenes simples. Sin embargo, son más complejos que una simple sucesión de fotografías, ya que la gran mayoría de ellos llevan incorporados un audio y un sentido espacio temporal.

A continuación se expondrán dichos estudios.

4.2.3.1. Trabajos y Estudios

El trabajo más interesante que se encuentra en este campo es el realizado por [MG18], que no sólo se embarca en la ambiciosa tarea de identificar la fuente de adquisición de un vídeo, sino que además investiga aquellos vídeos procedentes de una red social, en concreto en este caso, *Whatsapp*. [MG18] utilizan para sus experimentos los vídeos de 10 cámaras diferentes, empleando tres vídeos por cámara: uno grabado en interior, otro en exterior y el último es un vídeo de referencia que graba una superficie gris. A los dos primeros los llaman vídeos naturales y al último, vídeo plano.

Para la identificación de cada vídeo, se valen de la comparación de los patrones de PRNU de los vídeos. Para ello, utilizan el software *PRNUCompare version 2.2*, desarrollado por the Netherlands Forensic Institute. La idea es comparar los patrones PRNU de los vídeos naturales con el patrón PRNU del vídeo plano. Una vez resuelta la identificación de cada vídeo original, envían cada vídeo a través de *Whatsapp* de todas las combinaciones posibles, que son:

- De IOS a IOS.
- De IOS a Android.
- De Android a IOS.
- De Android a Android.

Una vez transmitidos, [MG18] extraen de nuevo los patrones PRNU y los comparan de forma similar a la comparación anterior.

Otro trabajo que vale la pena destacar en este ámbito es el de [YHW12], que propone un método de identificación utilizando los fotogramas extraídos de un vídeo. A partir de dichos fotogramas, extraen las características de probabilidad condicional (*Conditional Probability Features* (CPF)), que utilizarán para clasificar los vídeos, con la ayuda de una SVM. En concreto, se centran en identificar diferentes modelos de cámaras. Los vídeos utilizados en este trabajo pertenecen a cuatro cámaras convencionales diferentes, y tienen todos la misma duración: 9 segundos. Todos ellos han sido grabados con su tamaño y formato por defecto. Para los experimentos, se utilizaron tanto vídeos en movimiento como estáticos, utilizando 40 para entrenar y 8 para la validación.

4.2.3.2. Resultados y Propuestas

Tras realizar las comparaciones en el trabajo de [MG18], el propio algoritmo *PRNUCompare* realiza un estudio de la correlación que están obteniendo los resultados, el *Peak to Correlation Energy* o *Peak to Correlation Energy* (PCE). En todos los casos, todas las disparidades obtienen un PCE inferior a 100, mientras que los PCE de las correspondencias son mucho más altos (a excepción de uno). Esto quiere decir que los patrones de PRNU de los vídeos originales naturales tienen un mayor grado de similitud con el vídeo plano tomado con la misma cámara que con los vídeos planos filmados por otra cámara distinta.

Los resultados indican que es posible determinar la cámara fuente de un vídeo original, con una probabilidad muy alta de acierto. Los vídeos que provienen de una misma cámara muestran muchas semejanzas entre sí, lo que facilita clasificarlos entre ellos de otros de cámaras diferentes. Una vez han sido transmitidos por *Whatsapp*, esta precisión en la clasificación baja, aunque sigue siendo posible clasificar bien la gran mayoría de ellos, con unos resultados mucho más malos en el caso de IOS. Sin embargo, al pasar por esta red social, deja de ser posible determinar si dos vídeos tienen la misma cámara de origen.

[YHW12] extraen las características de probabilidad condicional de los fotogramas con valores de luminosidad de cada vídeo para calcular su eficacia en un dominio espacial en un primer experimento. La precisión que obtiene es de un 82.6 %, que es relativamente baja comparada con estudios de identificación de cámara de imágenes. En el segundo experimento, deciden usar el mismo conjunto de vídeos, tomando el valor de la luminosidad. Con estas características, el acierto fue total. Finalmente, en un último experimento, se tomaron vídeos más complejos con cambios en las escenas más apreciables, lo que les dio un resultado del 97.2 %.

Capítulo 5

Algoritmo Propuesto

En este capítulo, se procede a explicar la propuesta presentada en este trabajo: el desarrollo de un método para la identificación de la fuente de adquisición de vídeos basado en redes neuronales y específicamente utilizando una arquitectura de aprendizaje profundo como las redes neuronales convolucionales (CNN).

5.1. Descripción del Algoritmo

El objetivo principal de este trabajo es proponer un modelo de red neuronal convolucional capaz de identificar la cámara fuente de un vídeo que ha sido grabado con un dispositivo móvil, a partir de los fotogramas de dicho vídeo. A pesar de que se propone una única red como resultado final, se ha utilizado una segunda red como red inicial, en la cual se han probado los resultados para poder desarrollar la red final.

Ya que esta red trabaja con los fotogramas de un vídeo, inicialmente ha sido testeada con dos conjuntos de imágenes, uno procedente de cámaras tradicionales DSC y otro procedente de dispositivos móviles, para probar su eficiencia. El modelo de red convolucional obtenido se ha adaptado para funcionar con los fotogramas extraídos de los vídeos. Esta propuesta se compone de diferentes partes para poder obtener el resultado final: la identificación de la cámara que grabó el vídeo.

5.1.1. Extracción de los Fotogramas de un Vídeo

El primer paso de la propuesta es extraer un número de fotograma de un vídeo. El número de fotogramas a extraer dependerá de la duración de cada vídeo, por lo que no será el mismo en todos. Cada fotograma que se extraiga será guardado de la misma manera que los vídeos originales: agrupado primero por la marca del móvil o cámara que grabó el vídeo al que pertenece y en segundo lugar agrupado según el modelo. A cada fotograma se le otorgará la siguiente nomenclatura:

$$NombreVídeo - No.Frame - I\_.ExtensiónImagen$$

Para poder llevar a cabo bien la ejecución del algoritmo, será necesario que ningún nombre de los vídeos a identificar contenga guiones. En caso de contenerlos, deberá añadirse un símbolo distintivo y único entre el nombre del vídeo y el número del fotograma para poder agrupar más adelante los fotogramas según a qué vídeo pertenezcan.

5.1.2. Reducción de los Fotogramas en Áreas más Pequeñas

Si bien inicialmente se pensaba trabajar con cada fotograma en su resolución original, se propone dividir cada fotograma en 256 trozos (*patches*) de 64x64 píxeles para facilitar el trabajo de la red y mejorar su eficiencia. El proceso de creación de *patches* ha sido dividido en dos secciones:

La primera sección se encarga de encarga de asignar a cada modelo de cámara un número que será su clase. Este paso es muy sencillo ya que todos los fotogramas se encuentran agrupados primero según la marca de la cámara o móvil que grabó el vídeo, y luego según el modelo. También se recortan desde el centro todos los fotogramas para que sean todos del mismo tamaño. Toda esta información será guardada en un fichero *.npy* que será utilizado en la segunda sección de este proceso.

La segunda sección se encarga de la creación de los *patches* como tal, pero ésta se lleva a cabo con 12 procesos distintos, lo que puede provocar que el orden en el que se crean y guardan los *patches* no sea siempre el mismo. También, cabe destacar que tras la creación de los *patches*, se eliminan aquellos muy saturados, pues no aportan ningún tipo de información y perjudican a la precisión del análisis posterior.

5.1.3. Preparación de los Datos

En primer lugar, se comprime todos los *patches* que componen todos los fotogramas en un archivo *numpy* para agilizar la red neuronal convolucional propuesta. Seguidamamente, para facilitar la ejecución, se organiza los *patches* en una estructura de carpetas jerárquica, dónde el nivel más exterior sea la marca del dispositivo, seguido por el modelo y, finalmente, el nivel más inferior sean todos los *patches* pertenecientes a una marca y un modelo determinado.

En segundo lugar, una vez generado todos los *patches* de cada fotograma, se lleva a cabo una preparación antes de comenzar a entrenar la CNN. Esta preparación consiste en la generación de cinco archivos *.npy*. Dos de ellos contendrán los *patches* que entrenarán y validarán la red, respectivamente. Otros dos contendrán las clases a las que pertenece cada *patch*, siendo guardados en dos ficheros diferentes según si pertenecen al conjunto de entrenamiento o al de validación. El último archivo se llamará *class_patches.npy* y contendrá la información de a qué clase pertenece cada *patch* del conjunto de validación. En este archivo, cada *patch* está en el mismo orden que en el archivo *datasetest.npy*, lo que facilita su uso en la Subsección 5.1.4.

5.1.4. Clasificación de los *Patches* con la CNN Propuesta

Una vez hecho todo lo anterior, se utiliza la CNN propuesta para clasificar los fotogramas de cada vídeo, determinar a qué cámara pertenece cada uno y el porcentaje de precisión que presenta la CNN en su misión. En esta parte, el usuario podrá modificar los parámetros que desee de la red, siendo los que se proporcionan por defecto los que propone este trabajo. También podrá decidir si quiere simplemente validar un modelo que ya ha sido entrenado previamente o cuántos modelos quiere entrenar, para realizar la media de sus precisiones después. En el caso de sólo querer entrenar un único modelo, es posible guardarlo.

Para entrenar y validar la red, se utilizan los datos que se han guardado en los ficheros *.npy* en la Subsección anterior. Por cada modelo que se entrene, se realizará su validación y se guardarán las predicciones obtenidas en cada modelo i en un fichero llamado *predicts_vuelta %i.npy*, dónde i es el número del modelo entrenado. Estas predicciones que

se guardan son las predicciones que ha obtenido cada *patch* con ese modelo. Finalmente, también se guarda en otro fichero llamado *predicts_total.npy* la precisión total obtenida con cada modelo entrenado.

5.1.5. Cálculo de la Precisión de la CNN con Vídeos a partir de la Precisión Obtenida con los Fotogramas

Debido a que los fotogramas no representan una unidad por sí sola, su clasificación no es suficiente. Es necesario por tanto determinar a qué modelo pertenece cada vídeo en función de los resultados de sus fotogramas. Para ello, la última parte del algoritmo propuesto se encarga de determinar cuál es la cámara fuente de un vídeo, a partir de los resultados obtenidos por la CNN. Una vez determinada, se comparan los resultados con las clases a las que verdaderamente pertenecen dichos vídeos (información que ha sido guardada durante el proceso) para comprobar la eficacia del algoritmo propuesto.

Esto se lleva a cabo una vez obtenidos los ficheros *.npy* que contienen las predicciones de todos los *patches* con cada modelo, se lleva a cabo la *reconstrucción* de los vídeos a partir de esos *patches* para poder saber cuál es la predicción de la red con cada vídeo y por tanto, su precisión. Para ello, esta parte se divide en varias fases:

1. **Asociación de cada *patch* a un modelo.** Las predicciones que devuelve la CNN para cada *patch* tienen la forma de un array con tantos elementos como modelos de cámaras haya a clasificar. Cada elemento i del array será un número entre 0 y 1, que representará la probabilidad de dicho *patch* a pertenecer en la clase i. Para determinar a qué clase ha predicho la CNN que pertenece dicho *patch*, se toma la clase i tal que el elemento i del array es el máximo.

   Una vez determinado esto, se devuelve un array con todos los *patches*, la clase a la que pertenece y la clase a la que se ha predicho que pertence. Para conocer su clase real, se ha usado el archivo *class_patches.npy* que se guardó en la Subsección 5.1.3, y se basa en el hecho de que la CNN devuelve las predicciones en el mismo orden que se introdujeron los *patches* en la red.

2. **Agrupación de los *patches* según los fotogramas.** Gracias a la nomenclatura que se le dio a los *patches* a la hora de crearlos, en este punto se divide el nombre en dos de manera que se elimina la parte del nombre que contiene el número del *patch* y se agrupan los *patches* que tienen el mismo nombre. De esta manera, se tienen todos los *patches* que componen un fotograma agrupados. Una vez hecho esto, se decide la clase de cada fotograma, que será la clase que más se repita dentro de las clases de los *patches* que lo componen.

3. **Agrupación de los fotogramas según los vídeos.** Al igual que el paso anterior, se lleva un proceso muy similar con los fotogramas para agruparlos según los vídeos a los que pertenecen. Una vez agrupados los fotogramas de cada vídeo, se determina la clase que se ha predicho para ese vídeo eligiendo la clase que más se repita entre las clases de sus fotogramas.

   Por último, se evalúa la precisión del modelo evaluando cuántos vídeos han sido bien clasificados. Para ello, se compara la clase predicha con su clase real, que ha sido guardada a su vez en el array durante todo el proceso.

5.2. Desarrollo de la Red Propuesta

En la literatura disponible sobre el análisis de la identificación de las fuentes de adquisición, se puede observar que si bien existen varias investigaciones sobre la fuente de adquisición de imágenes, el número de publicaciones se reduce drásticamente al buscar investigaciones que estudien la fuente de adquisición de vídeos.

Por ello, animados por los resultados obtenidos en la literatura al aplicar las técnicas de Machine Learning a las imágenes para clasificarlas según la cámara que las capturó, en este trabajo se propone aplicar esas mismas técnicas a los vídeos. Como resultado, la propuesta de este trabajo es el uso de una red CNN para su clasificación.

Su elección se debe a ser una técnica muy poco estudiada en vídeos aún, y a que según los resultados de la literatura más actual, es uno de los métodos más rápidos y eficientes que existe hoy en día. Además, es un método capaz de adaptarse a distintos propósitos con facilidad.

5.2.1. Primera Red Neuronal Convolucional

La elección de las características de una red neuronal convolucional no son para nada triviales. La configuración de cada una está muy ligada a los tipos de datos con los que se trabajan, lo que puede dar lugar a redes neuronales convolucionales muy diferentes pero igualmente eficaces según sean el tamaño de los datos, la complejidad de las imágenes o incluso el objetivo final de dicha red.

Para empezar las pruebas que terminarían con la propuesta final, se comenzó con una red muy básica para poder ver sus virtudes y defectos e ir así puliéndola, con ayuda de los resultados obtenidos y de los consejos encontrados en la literatura.

Las características de esta red son:

1. **Tres capas convolucionales.** Esta primera red está compuesta por tres capas convolucionales, cada una de ellas inmediatamente seguida por una capa *Max Pooling* y un *dropout*, a excepción de la tercera, que prescinde de la capa *Max Pooling*.

   Los datos de cada capa aparecen representados en tablas: la Tabla 5.1 muestra los detalles de la primera capa convolucional de la Red 1, la Tabla 5.2 recoge los datos de la segunda capa convolucional de la Red 1 y la Tabla 5.3 se encarga de presentar la configuración de la tercera capa convolucional de la Red 1.

C. Convolucional 1			MaxPooling 1		Dropout 1
Nº Filtros	Kernel	Strides	Size	Strides	
28	5x5	1	2x2	1	0.2

Tabla 5.1: Detalles de la primera capa convolucional de la Red 1

C. Convolucional 2			MaxPooling 2		Dropout 2
Nº Filtros	Kernel	Strides	Size	Strides	
64	3x3	1	2x2	1	0.25

Tabla 5.2: Detalles de la segunda capa convolucional de la Red 1

C. Convolucional 3			Dropout 3
Nº Filtros	Kernel	Strides	
128	3x3	1	0.4

Tabla 5.3: Detalles de la tercera capa convolucional de la Red 1

2. **Dos capas completamente conectadas.** A continuación de las capas convolucionales, la Red 1 está compuesta por dos capas completamente conectadas, con las siguientes características que aparecen en la Tabla 5.4.

C. Completamente Con. 1			C. Completamente Con. 2	
Nº Filtros	Kernel	F. Activación	Nº Filtros	F. Activación
128	3x3	ReLU	Nº Clases a clasificar	Softmax

Tabla 5.4: Detalles de las capas completamente conectadas de la Red 1

3. **ReLU como función de activación.** Para comenzar los experimentos, se empezó probando con la función de activación no lineal más conocida, la función ReLU.

La Figura 5.1 muestra un dibujo esquemático de esta Red 1 que se ha descrito.

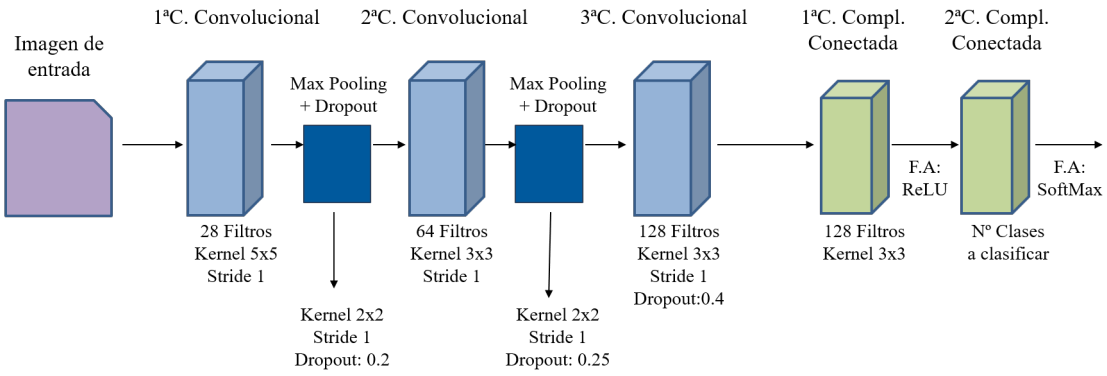

Figura 5.1: Esquema de la arquitectura de la Red 1.

5.2.2. Segunda Red Neuronal Convolucional

A raíz de la lectura de la propuesta de [FONBCS18] y de sus conclusiones para una red adaptada a imágenes tomadas por dispositivos móviles, se decidió incorporar parte de sus consejos en esta red, volviéndola similar a la suya. Se deseaba estudiar si sus buenos resultados se conservaban al aplicarse a vídeos en lugar de a fotografías. A partir de ahora, se referirá a esta red como Red 2 a lo largo del documento. Las características principales de esta red son:

1. **Kernel de las capas convolucionales.** Los kernel de estas capas se caracterizan por ser de una dimensión pequeña. En este caso concreto, los kernel de ambas capas serán de 3x3. Al ser de dimensión reducida, proporcionan mejor resultados frente a kernels de dimensión mayor.

2. **Dos capas convolucionales.** Las capas de una red neuronal convolucional cuyo objetivo es la identificación de la fuente de adquisición de una imagen, están centradas en identificar señales digitales que difieran de una cámara a otra. Según los estudios de [FONBCS18], estas señales son extremadamente sensibles, por lo que podrá *sobreaprender* rápidamente en casos en los que existan más de dos capas convolucionales, o quedarse corta en el aprendizaje si sólo existe una capa.

   Por ello, esta topología de dos capas convolucionales, que ha adoptado la Red 2, ha resultado ser la más eficaz independientemente del número de salidas, de las funciones de activación o de si existía *dropout* entre las capas. También se ha modificado el *stride* de uno a dos.

3. **ReLU como función de activación.** Para unos buenos resultados, era imprescindible aplicar una función de activación no lineal a la red. Hay dos grandes funciones recomendadas por todos los expertos para este tipo de objetivos: la ReLU y la Leaky ReLU. En este caso, al contrario que en los estudios de [FONBCS18], los mejores resultados fueron obtenidos con la función ReLU, si bien la diferencia fue mínima.

4. **Único *dropout* de 0.5** Tras varios experimentos, el mejor resultado obtenido con las redes ha sido utilizando este valor como único *dropout* a la salida de las capas convolucionales.

5. **Tres capas completamente conectadas** Se pudo comprobar que añadiendo una capa completamente conectada se obtenía una pequeña mejora en la clasificación final, por lo que se aumentó su número de dos a tres.

Los detalles de esta red aparecen recogidos en tablas, dónde la Tabla 5.5 muestra los detalles de las capas convolucionales de la Red 2 y la Tabla 5.6 representa los detalles de sus capas completamente conectadas.

C. Convolucional 1			C. Convolucional 2			MaxPooling		Dropout
Nº Filtros	Kernel	Strides	Nº Filtros	Kernel	Strides	Size	Strides	
28	3x3	2	64	3x3	2	3x3	2	0.5

Tabla 5.5: Detalles de las capas convolucionales de la Red 2

C. Completamente Con. 1		C. Completamente Con. 2		C. Completamente Con. 3	
Nº Filtros	F. Activación	Nº Filtros	F. Activación	Nº Filtros	F. Activación
256	ReLU	512	ReLU	Nº Clases a clasificar	Softmax

Tabla 5.6: Detalles de las capas completamente conectadas de la Red 2

La Figura 5.2 muestra un esquema de esta Red 2 que se ha descrito y que se presenta en este trabajo como propuesta.

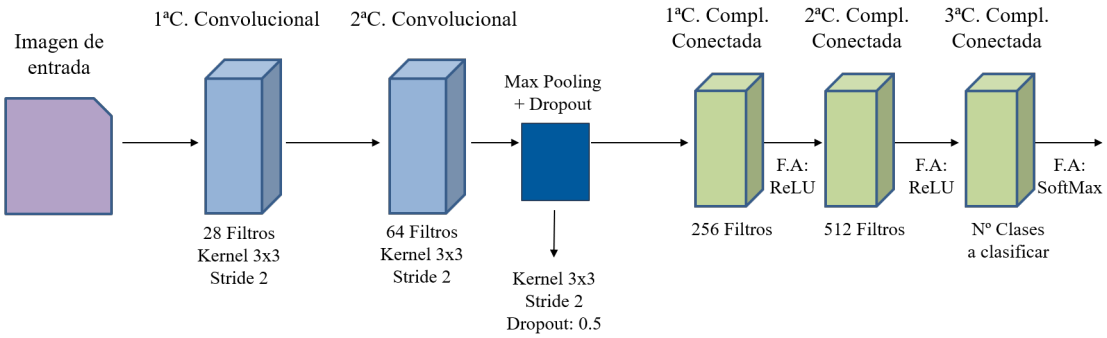

Figura 5.2: Esquema de la arquitectura propuesta para la Red 2.

5.3. Recursos y Herramientas Utilizadas

Todo el código que incorpora este trabajo como resultado a las investigaciones ha sido desarrollado en el lenguaje de programación *Python*. Para desarrollar e implementar la CNN, se han utilizado las librerías Keras y LibSVM, ya que facilitaban mucho esta tarea gracias a todas las funciones que ofrecen. El código de este trabajo está disponible en el enlace `http://gitlab.fdi.ucm.es/almudena/tfg_almudenaouteda.git`.

Todas las pruebas se han llevado a cabo en dos entornos de desarrollo:

- **PyCharm.** Se trata de un entorno de desarrollo integrado, utilizado en la programación de computadoras, específicamente para el lenguaje *Python*. La totalidad del código fue desarrollado y probado en esta plataforma, con la excepción de la CNN.
- **Google Colaboratory.** Esta herramienta es un entorno de máquinas virtuales basado en Jupyter Notebooks ofrecido por Google. Su único requisito es poseer una cuenta de correo de Google, y con ella es posible ejecutar código en la nube, es posible elegir correr un *notebook* en una CPU, GPU o en una TPU de forma gratuita.

  Este recurso fue utilizado por su disponibilidad de 12GB de memoria RAM y por ofrecer una GPU, lo que permitía paralelizar y acelerar enormemente el proceso de entrenamiento y evaluación de la CNN.

5.3.1. Configuración y Componentes de los Experimentos

Los experimentos se llevaron a cabo en un único ordenador. Los componentes y características de dicho ordenador son:

- **CPU**: Procesador Intel Core i5-5200U 4ª Gen. (Haswell) 2.20GHz.
- **Memoria RAM**: 8Gb.
- **Tarjeta gráfica**: AMD Mobility Radeon R5 M330.
- **Sistema Operativo**: Linux Ubuntu 18.04.

Capítulo 6

Experimentos y Resultados

En este capítulo se mostrarán y explicarán los experimentos llevados a cabo para probar las propuestas y los resultados obtenidos con cada uno de esos experimentos. Será estructurado de la siguiente manera: en primer lugar se detallarán los conjuntos elegidos para llevar a cabo los experimentos, a continuación se explicará el procesamiento de datos que se les ha hecho a dichos conjuntos para poder ser empleados en las redes y finalmente, se mostrarán y explicarán los resultados obtenidos tras el entrenamiento y la validación de cada conjunto con las Redes 1 y 2.

6.1. Elección de los Conjuntos de Muestras

Para poder llevar a cabo los experimentos, era necesario disponer de varios conjuntos de muestras que nos permitiesen entrenar y validar las redes propuestas. Se eligieron varios conjuntos de imágenes y un conjunto final de vídeos.

6.1.1. Conjuntos de Imágenes

Los primeros conjuntos elegidos fueron de imágenes, por dos motivos principales:

1. Como bien se ha comentado antes, las grandes propuestas han sido en su mayoría basadas en imágenes. Para poder tener un buen comienzo, fue preciso entender y replicar dichas propuestas para a partir de ellas, dar un salto adelante. Y para ello, eran necesarias las imágenes.

2. Un vídeo, de forma muy generalista, puede ser definido como un conjunto de imágenes sucesivas. Este puede tener incorporado o no un audio, que puede proporcionar o no una información extra que pueda ser de utilidad a la hora de identificar la fuente de adquisición. Por tanto, si la imagen es la 'unidad básica' de un vídeo, es necesario entenderlas primero antes de intentar ir más allá. En otras palabras, si la propuesta no funciona con imágenes, es muy poco probable que sea mejor con un vídeo.

Los conjuntos formados fueron los siguientes:

6.1.1.1. Dresden Dataset

El primer conjunto utilizado fue una parte del *dataset Dresden* [GB11]. Este *dataset* en su forma original cuenta con una recopilación de más de 16.000 imágenes de gran variedad en su contenido: pueden encontrarse fotografías tanto de interiores como de exteriores

en él. Estas imágenes han sido tomadas por 73 cámaras diferentes pertenecientes a 25 modelos distintos. Se trata de un *dataset* público y de acceso libre puesto a disposición de la comunidad científica y utilizado en varias investigaciones.

Para los experimentos, se seleccionó una pequeña parte de él: se tomaron todas las fotos realizadas por todos los dispositivos pertenecientes a cuatro modelos diferentes: Afga DC-504, Afga DC-733s, Canon Ixus55 y Nikon D200. Por comodidad, todos los datos de interés referentes a este conjunto de imágenes se muestran en la Tabla 6.1.

Marca	Modelo	Nº imgs. Entrenamiento	Nº imgs. Validación	Resolución Original
Afga	DC-504	51	21	4032x3024
Afga	DC-733s	106	45	3072x2304
Canon	Ixus55	126	53	2592x1944
Nikon	D200	416	178	3872x2592

Tabla 6.1: Información sobre el Conjunto 1

En este *dataset* se eligieron únicamente 4 modelos para probar la fiabilidad inicial de la propuesta a entrenar y testear, ya que cuántos menos modelos a clasificar, más fácil es en teoría su clasificación. Además, las investigaciones sobre las imágenes tomadas con cámaras de fotos, y en particular sobre este *dataset* son relativamente abundantes, luego no tenía mucho sentido profundizar con él. Para facilitar su nomenclatura y diferenciarlo del resto de conjuntos que se definen a continuación, este conjunto será llamado a partir de ahora el Conjunto 1.

6.1.2. Dataset Móviles GASS

Este segundo *dataset* ha sido creado por el Grupo de Análisis, Seguridad y Sistemas de la Universidad Complutense de Madrid con la finalidad de llevar a cabo varias investigaciones del departamento. El conjunto original consta de 11601 fotografías tomadas por 68 móviles diferentes pertenecientes a 48 modelos distintos. A partir de este *dataset*, se han elaborado dos conjuntos diferentes, donde el segundo es una ampliación del primero y siendo los dos una parte del total del *dataset* original.

El primer conjunto se ha formado con las fotografías tomadas por 5 dispositivos pertenecientes a 5 modelos diferentes, que son el LG p760, el Nokia 800-lumia, los Samsung S3 Neo y S4 Mini y el Xiaomi Mi3.

El motivo de la utilización de este conjunto, al que se llamará Conjunto 2, se debe a la intención de testear la propuesta con un conjunto de imágenes que no haya sido utilizado en otros experimentos o trabajos anteriormente y poder así verificar que esta red se adapta a cualquier conjunto y no sólo a los conjuntos ya 'predefinidos'. Además, era interesante utilizar un conjunto formado por imágenes tomadas con dispositivos móviles, ya que por la manufacturación de sus sensores, el hecho de disponer de dos cámaras un mismo dispositivo y el hábito que tienen varias marcas en compartir sensor, dificulta la tarea de identificar fielmente la fuente de adquisición usada. Para comenzar, se eligió un conjunto formado por 5 únicas clases para facilitar la identificación y comprobar su buen funcionamiento.

El segundo conjunto, al que se llamará Conjunto 3, es simplemente una ampliación del Conjunto 2. A dicho conjunto se le han añadido las fotografías tomadas por los dispositivos pertenecientes a 5 modelos más, para tratar de dificultar el proceso de identificación y valorar así su eficiencia. Los datos se reflejan en la Tabla 6.2.

Marca	Modelo	Nº imgs. Entrenamiento	Nº imgs. Validación	Resolución Original
LG	p760	100	27	1944x2592
Nokia	800-lumia	100	27	3264x2448
Samsung	S3 Neo	50	27	1280x720
Samsung	S4 Mini	50	27	3264x1836
Xiaomi	Mi3	100	27	4208x2364
Aquaris	M5	80	20	4096x2304
Apple	iPhone 6 Plus	100	100	3264x2448
OnePlus	One A0001	100	100	4160x3120
Oukitel	K6000 Pro	50	50	3120x4160
Xperia	M2_D2303	100	100	3104x1746

Tabla 6.2: Información sobre el Conjunto 2 y 3

6.1.3. Conjuntos de Vídeos

Para trabajar con vídeos, solo se ha escogido un único conjunto, al que se referirá a lo largo del trabajo como Conjunto de Vídeos. Este conjunto está compuesto por vídeos grabados por 5 modelos de dispositivos diferentes, también obtenidos del *dataset* del Grupo de Análisis, Seguridad y Sistemas de la Universidad Complutense de Madrid. Los detalles se pueden observar en la Tabla 6.3.

Marca	Modelo	Nº Dispositivos	Nº Vídeos	Resolución Original
Apple	iPhone 6	2	5	1920x1080
OnePlus	One A0001	1	5	1920x1080
Samsung	Galaxy S3	1	5	1920x1080
Xiaomi	Mi3	1	5	1280x720
Xperia	M2_D2303	1	5	1920x1080

Tabla 6.3: Información sobre el Conjunto de Vídeos

6.2. Tratamiento y Preprocesamiento de los Conjuntos

Los conjuntos han sido sometidos a distintos preprocesamientos para poder ser utilizados en los experimentos.

6.2.1. Selección y Creación de los Conjuntos

Ninguno de los conjuntos utilizados en estos experimentos son el total de un *dataset* ya existente, sino que son una parte de uno de ellos. El primer paso de los preprocesamientos fue, por tanto, crear esos subconjuntos de los *dataset* ya existentes. Se trató simplemente de elegir dentro de todos los modelos disponibles de cada *dataset* un número de modelos fijo y utilizar todas las imágenes disponibles de esos modelos. Esto fue así en los Conjuntos 1, 2 y 3.

Para el Conjunto de Vídeos, en cambio, se seleccionó también un número fijo de vídeos (6) de cada modelo, en vez de usar todos los disponibles.

6.2.2. Separación en Dos Conjuntos: Entrenamiento y Validación

Este tratamiento fue solamente llevado a cabo con el Conjunto 1 y el Conjunto de Vídeos.

El Conjunto 1 en su forma original se encontraba organizado en una estructura de carpetas dónde cada una era una marca de cámaras fotográficas, y en las que a su vez, en su interior se encontraban más carpetas en las que cada una era un modelo de dicha marca. Finalmente, en el interior de las carpetas de los modelos se encontraban las fotografías originales pertenecientes a cada modelo. Tras seleccionar los modelos a utilizar, fue necesario dividir las imágenes de cada modelo en dos grupos: uno para entrenamiento y otro para validación. Para ello, se contó el número de imágenes disponibles de cada modelo y se dividieron de forma aleatoria: el 70 % de las imágenes de cada modelo fueron para la carpeta 'Entrenamiento' y el 30 % restante, para la de 'Validación'.

La división el el Conjunto de Vídeos fue mucho más simple: de los 6 vídeos seleccionados, se dejaron 5 para el conjunto de entrenamiento y se apartó uno para que formase parte del conjunto de validación.

Con los Conjuntos 2 y 3 no fue necesario este tratamiento ya que la organización original de sus conjuntos *padre* tenía esta división ya realizada.

6.2.3. Extracción de los Fotogramas de cada Vídeo

En este caso, lo primero de todo para poder preparar el conjunto de datos de entrenaniento de la CNN, es necesario extraer los fotogramas de cada vídeo. Por cada vídeo, se extrajo una cantidad variable de fotogramas o fotogramas de cada vídeo y se guardaron juntos todos los fotogramas pertenecientes a un modelo, es decir, cada carpeta de modelo contenía los fotogramas de todos los vídeos pertenecientes a dicho modelo. El número de fotogramas extraídos por vídeo es proporcional a la duración del vídeo, por lo que los vídeos más extensos disponían de más fotogramas. Para evitar su mezcla y saber qué fotograma pertenece a qué vídeo, cada fotograma adoptó la siguiente estructura en su nombre:

$$NombreVídeo - N^{o}Fotograma - I\_.ExtensiónImagen$$

Una vez extraídos todos, nos queda un conjunto nuevo, al cuál nos referiremos como el Conjunto 4. Sus datos más relevantes se presentan en la Tabla 6.4.

Marca	Modelo	Nº Imgs. Entrenamiento	Nº Imgs. Validación	Resolución Original
Apple	iPhone 6	101	96	1920x1080
OnePlus	One A0001	53	14	1920x1080
Samsung	Galaxy S3	55	11	1920x1080
Xiaomi	Mi3	185	32	1280x720
Xperia	M2_D2303	417	59	1920x1080

Tabla 6.4: Información sobre el Conjunto 4

6.2.4. División de cada Imagen en Áreas más Pequeñas

Inicialmente, se trató de usar las imágenes de cada conjunto con su resolución original, pero no fue viable por su gran tamaño, el elevado número de imágenes de las que se disponía y por los recursos de memoria con los que se contaba.

Por ello, siguiendo los consejos de [BLG+17], se desarrolló un algoritmo a partir del código de los experimentos de estos investigadores que permitía dividir cada imagen en pequeños trozos o *patches* que no solapasen entre sí. Era posible elegir el tamaño de cada *patch* y el número total de ellos que se quería por imagen. Se comenzó dividiendo cada imagen en 32 *patches* de 64x64 píxeles, pero como se verá en los apartados siguientes, estos valores se fueron modificando poco a poco hasta llegar a los ideales.

6.3. Resultados Obtenidos con la Red 1

Una vez la primera red estuvo definida y desarrollada, fue probada su eficiencia con los diferentes conjuntos antes descritos. Estos fueron los resultados.

6.3.1. Resultados del Conjunto 1 con la Red 1

Los primeros resultados obtenidos con la Red 1 fueron bastante aceptables a pesar de las malas condiciones en las que fueron ejecutados. Por el equipo del que se disponía, ejecutar 20 vueltas o *epochs* del Conjunto 1 podía llegar a tardar del orden de 30 minutos. Para poder realizar los cálculos, era necesario ejecutar cada entrenamiento 10 veces, con cuyos resultados se realizaba una media después.

En la Tabla 6.5 se muestran los resultados obtenidos en estos experimentos. El significado de las cabeceras de cada columna es el siguiente:

- **A:** Conjunto de datos empleado en los experimentos.
- **B:** Nº de imágenes del conjunto usadas para entrenamiento.
- **C:** Nº de imágenes del conjunto usadas para validación.
- **D:** Nº de áreas o *patches* en los que está recortada cada imagen.
- **E:** Dimensión de cada *patch*.
- **F:** Tamaño de cada *batch*.
- **H:** Precisión obtenida en el experimento en la red con los datos anteriores.

La columna *H* muestra la media final obtenida tras evaluar 10 veces la red con los mismos parámetros. En los experimentos de esta tabla, cada imagen del Conjunto 1 fue dividida en 32 *patches* de 64x64 píxeles, pues era el tamaño recomendado en un primer momento por [BLG+17].

A	B	C	D	E	F	G	H
Conjunto 1	25526	6122	32	64x64x1	10	20	82.778 %
Conjunto 1	25526	6122	32	64x64x1	10	30	82.473 %
Conjunto 1	25526	6122	32	64x64x1	10	50	**83.585 %**
Conjunto 1	25526	6122	32	64x64x1	10	70	82.866 %
Conjunto 1	25526	6122	32	64x64x1	10	100	81.813 %

Tabla 6.5: Resultados de la Red 1 con el Conjunto 1

En este primer momento el interés se centró en las *epochs*, ya que era necesario saber cuál era el número ideal de *epochs* con las que obtener los mejores resultados, pues no se contaba con excesivos medios y así sería posible optimizar mejor el tiempo y la investigación. Como se ha resaltado en negrita en la tabla, el mejor resultado obtenido fue con 50 *epochs*, consiguiendo una precisión del 83.585 %. Esta precisión, sin ser brillante, no era mala del todo para un primer intento.

El siguiente testeo para mejorar la red fue el *batch size*. Tomando iguales todos los parámetros de la tabla anterior y fijando las *epochs* a 50, se procedió a estudiar el *batch size* con la esperanza de obtener una mejora. Los resultados pueden verse en la Tabla 6.6. La relación del significado de las cabeceras de cada columna es la misma que la de la Tabla 6.5.

A	B	C	D	E	F	G	H
Conjunto 1	25526	6122	32	64x64x1	10	50	83.585 %
Conjunto 1	25526	6122	32	64x64x1	30	50	85.323 %
Conjunto 1	25526	6122	32	64x64x1	50	50	84.978 %
Conjunto 1	25526	6122	32	64x64x1	-	50	**86.403 %**

Tabla 6.6: Resultados de la Red 1 con el Conjunto 1

Aquí, los mejores resultados se obtuvieron tras no especificar el *batch size*, y mejoraron en casi un 3 % con respecto a los anteriores. Parte de esta mejora se debió a la posibilidad de obtener mejores recursos de computación, como una RAM más grande. Esto permitió poder experimentar más y ampliar la dimensión del *batch* hasta obtener unos resultados mejores.

6.3.2. Resultados del Conjunto 2 con la Red 1

Tras los resultados obtenidos con el Conjunto 1, se probó la eficiencia de la Red 1 en el Conjunto 2. A pesar de contar con la capacidad de cómputo mejorada que se había conseguido en los últimos experimentos y de emplear la misma configuración para la Red 1 que en el apartado anterior, los resultados que se obtuvieron aquí no fueron buenos, como pueden verse en la Tabla 6.7. La relación del significado de las cabeceras de cada columna es la misma que la de la Tabla 6.5.

A	B	C	D	E	F	G	H
Conjunto 2	12800	4320	32	64x64x1	10	50	30.385 %
Conjunto 2	12800	4320	32	64x64x1	-	100	**33.71 %**

Tabla 6.7: Resultados de la Red 1 con el Conjunto 2

Ante tan malos resultados, se procedió mejorar la red y probar una nueva: la Red 2.

6.4. Resultados Obtenidos con la Red 2

La Red 2 y los resultados obtenidos con ella revelan dos cosas:

1. El hecho de cambiar y afinar la red proporcionó mejores resultados en la clasificación, lo cual señala que tener una buena red es un punto muy importante.

2. Ya no sólo los cambios en la red proporcionan una mejoría, sino también el jugar y afinar el preprocesamiento de datos, como se pudo ver en esta fase de la investigación.

A continuación, aparecen los resultados representados en tablas.

6.4.1. Resultados del Conjunto 1 con la Red 2

Ya que los mejores datos obtenidos hasta ahora fueron con el Conjunto 1, y para mantener el mismo orden que en las pruebas anteriores, se entrenó y testeó la Red 2 con este conjunto primero. Los resultados están reflejados en la Tabla 6.8. La relación del significado de las cabeceras de cada columna es la misma que la de la Tabla 6.5.

A	B	C	D	E	F	G	H
Conjunto 1	25526	6122	32	64x64x1	-	50	92.496 %
Conjunto 1	25526	6122	32	64x64x1	100	300	**94.325 %**
Conjunto 1	25526	6122	32	64x64x1	-	300	94.315 %

Tabla 6.8: Resultados de la Red 2 con el Conjunto 1

Como puede apreciarse, los resultados mejoraron hasta en un 8 %, logrando una precisión realmente buena en el mejor de los casos y probando la mejora de esta red frente a su predecesora la Red 1.

6.4.2. Resultados del Conjunto 2 con la Red 2

Tras los buenos resultados con el Conjunto 1, era necesario contestar a la siguiente pregunta: ¿Sería capaz esta red de mejorar también los resultados del Conjunto 2?

A	B	C	D	E	F	G	H
Conjunto 2	12800	4320	32	64x64x1	-	100	41.824 %
Conjunto 2	12800	4320	32	64x64x1	-	300	41.755 %
Conjunto 2	12800	4320	32	64x64x1	100	300	**43.131 %**
Conjunto 2	12800	4320	32	64x64x1	50	300	41.535 %
Conjunto 2	12800	4320	32	64x64x1	200	300	42.646 %

Tabla 6.9: Resultados de la Red 2 con el Conjunto 2

Como puede apreciarse en la Tabla 6.9, los resultados mejoraron hasta un 10 %, pero no sólo estaban lejísimos de los resultados obtenidos con el conjunto 1, sino que además seguían por debajo del 50 % de precisión, lo que significaba que la red fallaba más de lo que acertaba en su clasificación. La relación del significado de las cabeceras de cada columna es la misma que la de la Tabla 6.5.

¿A qué se debían una diferencia tan abismal de resultados? A priori, parecía que podía estar causado por dos motivos:

1. El primer motivo podría estar relacionado con el sensor que capturó las imágenes. Esto significaría que las imágenes tomadas con un móvil pudiesen tener menos información en sus metadatos o más confusa, lo que propiciase que la red no fuese tan precisa en su clasificación.

2. El segundo motivo tendría que ver con el tamaño de la muestra del Conjunto 2. Como puede verse a lo largo de las tablas presentadas, el número de imágenes del Conjunto 2 es aproximadamente la mitad que las que tiene el Conjunto 1. Por tanto, esta falta de imágenes en la muestra, podría estar perjudicando a los resultados ya que la red no es capaz de aprender lo suficiente.

Tras este análisis, se intentó encontrar un paliativo al segundo motivo, la falta de imágenes. Dado que en un futuro próximo no era posible aumentar ese *dataset*, se optó por aumentar el número de *patches* por foto. Los resultados están en la Tabla 6.10. La relación del significado de las cabeceras de cada columna es la misma que la de la Tabla 6.5.

A	B	C	D	E	F	G	H
Conjunto 2	100312	33444	256	64x64x1	100	300	**87.205 %**
Conjunto 2	100312	33444	256	64x64x1	-	300	86.388 %
Conjunto 2	102400	34560	256	32x32x1	100	300	74.238 %
Conjunto 2	102400	34560	256	32x32x1	-	300	73.402 %

Tabla 6.10: Resultados de la Red 2 con el Conjunto 2

Se realizaron varias pruebas: se aumentó el número de *patches* a 256 en todos los casos, y a su vez se investigó cómo afectaba la dimensión de dichos *patches* y de los *batches*. Los resultados fueron incluso mejores de lo previsto, ya que se logró una mejora del 44 % en la precisión, llegando a un 87.205 % en el mejor de los casos. Si bien aún estaban lejos del Conjunto 1, esto era una mejora muy importante de cara a seguir con las investigaciones.

6.4.3. Resultados del Conjunto 3 con la Red 2

Llegados a este punto, se aumentó el Conjunto 2 añadiendo nuevas imágenes y modelos, creando como resultado el Conjunto 3, anteriormente definido. Se presentaban ahora dos objetivos: comprobar que la mejora de precisión lograda en el Conjunto 2 se mantuviese en el Conjunto 3, y tratar de encontrar el mejor preprocesamiento y configuración de la red para lograr la mayor precisión posible.

A	B	C	D	E	F	G	H
Conjunto 3	26528	16160	32	64x64x1	100	300	41.313 %
Conjunto 3	106240	64582	128	64x64x1	100	300	61.222 %
Conjunto 3	210392	127978	256	64x64x1	100	300	**68.568 %**
Conjunto 3	106240	64384	128	32x32x1	100	300	47.88 %
Conjunto 3	212480	129280	256	32x32x1	100	300	54.34 %

Tabla 6.11: Resultados de la Red 2 con el Conjunto 3

Aplicando los mismos razonamientos que con el Conjunto 2, se aumentó el número de *patches* de las imágenes del Conjunto 3. Si bien no se consiguió alcanzar el buen resultado del Conjunto 2, se logró una precisión de un poco menos del 70 %, lo cuál no está del todo mal. Estos resultados se encuentran en la Tabla 6.11. La relación del significado de las cabeceras de cada columna es la misma que la de la Tabla 6.5.

6.4.4. Resultados del Conjunto 4 con la Red 2

Para poder entrenar y testear la Red 2 con el Conjunto 4, era necesario en primer lugar, extraer los fotogramas de cada vídeo como se ha descrito anteriormente. Una vez obtenidos, la red era usada como en los casos anteriores: primero se ha troceado cada fotograma en 256 *patches* de 64x64 píxeles cada uno y se han utilizado para entrenar y testear la red.

El problema final estaba al llegar a los resultados: la red devolvía la clase a la que pertenecía cada *patch* de los fotogramas, cuando el objetivo que se buscaba era saber a qué clase pertenece cada vídeo como un total. Para lograr esto, se llevó a cabo el diseño e implementación de un algoritmo auxiliar, con el objetivo de concluir cuál era la clase de cada vídeo, tomando el vídeo como elemento único.

6.4.4.1. La Clase como la Moda de los Resultados de los Fotogramas

Una vez obtenidos los resultados de las predicciones de la red para cada *patch*, se desarrolló un algoritmo que fuese capaz de hacer los siguientes pasos:

1. Como primer paso, el algoritmo se encarga de determinar cuál es la clase a la que pertenece cada predicción obtenida, pues los datos de los que se disponen son una lista que contiene una lista con tantos elementos como clases a clasificar, y en el interior de cada lista de clase hay tantas listas como *patches* pertenecientes a esa clase. Cada lista de *patch* contiene los porcentajes de clasificación de ese *patch* a cada una de las clases.

   Para asignar la clase a la que pertenece cada *patch*, se ha tomado la clase en la que el porcentaje de clasificación es el mayor. Como resultado, obtenemos una lista con tantas listas como clases, donde cada lista de clase tiene tantos elementos como *patches*. Cada elemento contiene la clase a la que pertenece ese *patch* según los resultados obtenidos en la red y el porcentaje asociado.

2. La segunda función ha consistido en asociar los resultados de los *patches* obtenidos en el paso anterior a cada *patch* conocido. Para ello, se ha usado el conocimiento de que cada resultado fue devuelto en el orden en el que se introdujeron los *patches* y una lista auxiliar que contiene el orden de cada *patch*, con su clase y su nombre conocido para poder identificar cada uno.

   El objetivo de esto es saber qué *patch* pertenece a cada fotograma y poder así agruparlos, primero por clases, y luego por fotogramas. Una vez agrupados, se le ha asociado a cada fotograma la clase que más frecuencia tuviese entre los *patches* que la formasen. En caso de tener algún empate, se seleccionaba la clase que tuviese el porcentaje más alto entre las que tuviesen mayor frecuencia.

   Como resultado, se obtenía una lista con tantas listas como clases. Cada lista de clase contenía a su vez tantos elementos como fotogramas perteneciesen a esa clase, junto con la clase a la que deberían pertenecer y la clase en la que ha sido clasificado.

3. El penúltimo paso ha consistido en realizar un proceso similar para agrupar cada fotograma según el vídeo al cuál pertenece. Una vez agrupados, primero por clase, y luego por vídeos, se ha procedido a calcular la clase a la que pertenece cada vídeo, calculando la clase más repetida de sus fotogramas. En este caso, en caso de darse un empate, se seleccionaba al azar una de las clases de la moda múltiple.

   El resultado obtenido ha sido una lista con tantas listas como clases, donde cada lista de clase tenía tantos elementos como vídeos. Cada vídeo contenía la información

de su nombre, la clase a la que debía pertenecer y la clase que le ha otorgado la clasificación.

4. Para finalizar, se ha calculado la precisión de la clasificación calculando el porcentaje de vídeos bien clasificados.

Los resultados finales aparecen en la Tabla 6.12. Además de aparecer las mismas columnas que las tablas anteriores, se han añadido un par de columnas nuevas. La relación del significado de las cabeceras de las columnas de la A a la G es la misma que la de la Tabla 6.5. El significado de la columnas nuevas es el siguiente:

- **H:** Precisión obtenida en la clasificación de los *patches* que forman cada fotograma. Al igual que en las tablas anteriores, la precisión que se muestra aquí es el resultado de hacer la media de las precisiones obtenidas en 10 ejecuciones distintas.
- **I:** Precisión obtenida en la clasificación de los vídeos. Se ha obtenido de forma similar a la anterior, pero con ligeras diferencias: Primero, se han calculado los resultados obtenidos para los vídeos 4 veces para cada resultado obtenido con los fotogramas y se ha hecho la media de ellos. A continuación, se ha hecho la media de las medias obtenidas en el paso anterior. Esta última media es la que se refleja en la tabla.

A	B	C	D	E	F	G	H	I
Conjunto 4	203004	51072	256	64x64x1	100	20	80.13 %	**89 %**

Tabla 6.12: Resultados de la Red 2 con el Conjunto 4

Por último, la Tabla 6.13 muestra la matriz de confusión con los resultados obtenidos. Como puede apreciarse, los únicos modelos que presenta dificultades en su clasificación son el OnePlus One_A0001, que es confundido con el Xiaomi Mi3 y el Samsung Galaxy S3, que se confunde mínimamente con el OnePlus One_A0001.

	Apple iPhone 6	OnePlus One_A0001	Samsung Galaxy Galaxy S3	Xiaomi Mi3	Xperia M2_D2303
Apple iPhone 6	30	0	0	0	0
OnePlus One_A0001	0	18	0	12	0
Samsung Galaxy S3	0	1	29	0	0
Xiaomi Mi3	0	0	0	30	0
Xperia M2_D2303	0	0	0	0	30

Tabla 6.13: Matriz de confusión del Conjunto 4 con la Red 2

Capítulo 7

Conclusiones y Trabajo Futuro

En este último capítulo se recogerán las conclusiones a las que se ha llegado después del desarrollo de la propuesta de este trabajo y los numerosos experimentos llevados a cabo. A continuación, también se detallarán posibles trabajos futuros que puedan continuar esta línea de investigación, ampliarla y mejorarla.

7.1. Conclusiones

Una vez finalizado este trabajo se han obtenido las siguientes conclusiones:

En la literatura disponible actualmente, hay una escasez muy grande en la investigación de la fuente de adquisición de los vídeos. Como se ha explicado anteriormente en la introducción, esto es un problema muy grande, ya que el auge de las técnicas que son capaces de manipular los vídeos dificulta en gran medida poder fiarse del contenido que presentan los mismos y de su fuente de procedencia. Este problema se está solucionando poco a poco con las imágenes, pero no así con los vídeos, que llevan un avance mucho más lento.

Como se ha podido ver en el capítulo de Experimentos y Resultados, se ha obtenido una precisión del 89 % en la clasificación de vídeos según la cámara que los grabó con la red neuronal convolucional propuesta en este trabajo. Estos resultados, aún estando por debajo del 90 % y siendo inferiores a la precisión obtenida trabajando con imágenes, se pueden considerar muy buenos. Al afirmar que son buenos resultados, se ha tenido en cuenta también que, al haber muy pocas publicaciones que utilicen una CNN para identificar la fuente de adquisición de un vídeo, esta precisión obtenida es un resultado más que decente y aceptable para tratarse de una de las primeras aproximaciones que existen a la identificación de la cámara fuente de un vídeo con este tipo de método para clasificación.

Para llevar a cabo los experimentos presentados en este trabajo, se ha utilizado un ordenador personal con una RAM de 8GB que, a priori, parecía suficiente. Sin embargo, es necesario destacar que debido al gran número de imágenes utilizadas y a su gran resolución, pronto se quedó muy corta, ralentizando enormemente las ejecuciones (llegando a tardar en torno a 3h cada una) o incluso llegándose a saturar e impidiendo su ejecución. Este problema fue paliado en parte con el uso de la herramienta Google Colaboratory, que cuenta con hasta 12GB de memoria RAM. Sin embargo, a la hora de ejecutar los experimentos con el Conjunto 4, se hizo patente de nuevo la escasez de memoria, por lo que se sugiere que una mejora en los recursos utilizados podría representar una ejecución más ágil y unos mejores resultados.

En este trabajo, como se explica en capítulos anteriores, a la hora de alimentar la red, se dividen primero todas las imágenes en áreas más pequeñas o *patches*. Esto es así, por dos motivos: por la imposibilidad de ejecutar el código con las imágenes en su resolución original por falta de memoria, y por las sugerencias de [FONBCS18, BLG+17] de dividir así las imágenes. Se quiere señalar que, al contrario de lo que sugiere [FONBCS18], los mejores resultados se han obtenido al dividir cada imagen en trozos de 64x64 píxeles, no de 32x32 como sugieren ellos. Además, ha sido necesario elevar el número de *patches* por fotos de 32 que propone [BLG+17] a 256, obteniendo la mejor precisión con esta cifra.

El algoritmo presentado en este trabajo está diseñado de tal forma que presenta una gran versatilidad, permitiéndole realizar que lleva a cabo un par de tareas más diferentes de la original:

En primer lugar, su estructura le permite ser capaz de analizar la fuente de adquisición de imágenes además de las de los vídeos. Para ello, bastaría con comentar en el código la parte del algoritmo destinada a la extracción de los fotogramas y la parte final en la que se encuentra la clase de cada vídeo a partir de dichos fotogramas. Habría que asegurarse también de que la carpeta dónde se encuentran las fotografías a analizar tiene la misma estructura que la carpeta original FRAMES.

En segundo lugar, el algoritmo permite analizar la precisión de un modelo previamente entrenado y generado, sin necesidad de utilizar la CNN propuesta en el código y en este trabajo.

7.2. Trabajo Futuro

Como posibles trabajos futuros pueden señalarse los siguientes:

- **Comparación de este Trabajo con una SVM.** Se propone como una continuación lógica de este trabajo un estudio sobre la identificación de la fuente de adquisición de vídeos usando una SVM. Se cree que puede ser una buena idea desarrollar una SVM capaz de identificar la cámara que grabó un determinado vídeo a partir de sus fotogramas, igual que se ha hecho aquí, con el fin de poder comparar los resultados obtenidos con uno y otro método de clasificación. Esto permitiría ver cuál de los dos obtiene una precisión más alta, si es posible mejorar su tiempo de ejecución o si se podría llevar a cabo con unos recursos menos exigentes de los que necesita la red neuronal convolucional. Los resultados que se arrojasen de aquí podrían determinar cuál es el método más idóneo para clasificar vídeos, y facilitar así el aumento de estudios y resultados al respecto, ya que es un campo poco estudiado.

- **Utilización de algún Sistema Tipo SHAP o LIME.** El poder explicar un modelo es una prioridad hoy en día para la comunidad científica, ya que se evitarían los sesgos de los modelos y permitiría a los responsables de las decisiones comprender cómo funcionan para utilizarlos de la mejor manera posible. SHAP y LIME son dos librerías del lenguaje de programación *Python* cuyo objetivo es intentar explicar el modelo de una CNN. SHAP se encarga de considerar todos las predicciones posibles de una instancia a partir de todos sus posibles valores de entrada, y gracias a este enfoque exhaustivo, puede garantizar propiedades como la consistencia y la precisión local. LIME construye modelos lineales dispersos de cada predicción para explicar su funcionamiento. La incorporación de estas librerías a este trabajo podrían aportar información interesante y útil de cara a mejorar los resultados presentados y aumentar la precisión ya obtenida.

- **Incorporación del Audio del Vídeo.** En este trabajo, para identificar la fuente de adquisición de un vídeo se utiliza únicamente la información contenida en sus fotogramas, extraídos previamente. Esto significa dos cosas: En primer lugar, significa que toda la información que pueda contener el audio, se pierde. Al ser desechado el audio, no se tiene en cuenta si aporta información valiosa que puede ayudar a mejorar la clasificación del vídeo o si bien al revés, puede perjudicar dicha clasificación. Se propone como trabajo futuro la incorporación del audio a la clasificación, y la comparación de estos resultados con los que aquí se presentan. El segundo significado es que puede ser que en un vídeo el único elemento que sea falsificado sea su audio. Si esto es así, el estudio de su fuente de adquisición a través únicamente de los fotogramas es inútil, ya que no detectará ninguna modificación. Se cree necesario por tanto, el desarrollo de un estudio adicional en el que se evalúe la fuente de adquisición del vídeo a partir de su audio, o que se añada a este dicha información. De esta manera, se complementará el trabajo y es posible que se reduzca el ratio de error de este trabajo.

Capítulo 8

Introduction

8.1. Motivation

Presently, the vast majority of the population carries a mobile phone in their pocket, and even more than one. In the beginning, these devices had very basic functions, such as making or receiving calls or being able to send and receive messages. Little by little, they have increased their functionalities, by adding more features such as games, calendars and mobile cameras. Today, the majority of mobiles have, at least, one camera, which is able of taking photographs as well as videos. This, added to the simple use of these devices, makes the action of taking a photograph or video at any time and in any place a normalized act in the day to day of any person.

Frequently, until a few years ago, the contents which were shown in a video or image, were considered as an incuestionable truth. However, there has been a rise of tampering methods and technologies which allow to modify these contents, even for these people who has no knowledges about the area [GKWB07]. This phenomenon has implied that it is no more possible to trust in a photograph or video. Therefore, while the number of scams in the images increases, a series of studies and investigations have been developed in parallel, that are able to determine univocally whether an image or a video has been altered or shows its original content.

One of the most important approaches that has been given to these investigations and advances is the attempt to discover which camera was the one that recorded a certain video or image. This is done from the video or image itself, with no more help than the information it may contain. Fortunately, at the time of recording, each camera leaves in the captured elements an own footprint with different characteristics, which allows to track those characteristics and differentiate one camera from another.

Advances in this field are different according to whether they are images or videos, or if they were recorded with a traditional camera or with the camera of a mobile device, but all of them are very important to improve these identification techniques and be useful to society. Thanks to these works, it is possible to determine if a certain video or photograph can be used as judicial evidence and to trust without any doubt in the content that is shown. This has facilitated the admission of images and videos from various cameras, especially security camcorders as judicial evidence, and has helped to identify the perpetrators based on the identification of the device that recorded the images.

8.2. Context

This End of Degree Paper is part of a research project entitled RAMSES, approved by the European Commission within the Horizon 2020 Research and Innovation Framework Programme (H2020-FCT-2015, Innovation Action, Proposal Number: 700326) and in which the GASS Group of the Software Engineering and Artificial Intelligence Department participates, integrated in the Faculty of Computer Science of the Complutense University of Madrid (Analysis, Security and Systems Group, http://gass.ucm.es, group 910623 of the catalogue of research groups recognised by the UCM).

In addition to the Complutense University of Madrid, the following participate entities:

- Treelogic Telematics and Rational Logic for Empresa Europea SL (Spain)
- Ministério da Justiça (Portugal)
- University of Kent (United Kingdom)
- Centro Ricerche e Studi su Sicurezza e Criminalità (Italy)
- Fachhochschule fur Offentliche Verwaltung und Rechtspflege in Bayern (Germany)
- Trilateral Research & Consulting LLP (United Kingdom)
- Politecnico di Milano (Italy)
- Service Public Federal Interieur (Belgium)
- Universität des Saarlandes (Germany)
- Dirección General de Policía - Ministerio del Interior (Spain)

8.3. Object of the Investigation

It has been shown that it is possible to identify the source of acquisition of an image or a video through the characteristics of its internal elements and its metadata. These characteristics, which are acquired during the creation process and in the subsequent processing, are part of what could be considered the DNA of an image or video, and being analyzed, they can show decisive information about a digital content. Part of these signs of identity are granted by the source camera that took the image or video, which leaves its own mark on their creations, and which will allow it to be identified a posteriori.

Most of the research conducted in recent years on source identification techniques has focused on the identification of traditional cameras DSC. With the appearance and massification of mobile devices, mostly capable of capturing photographs, it was considered necessary to carry out new research on techniques to identify the source of images generated by mobile devices. The vast majority of these studies have been done focusing on images, so there are only a few which looked at videos.

The present investigation focuses on the techniques of identifying a video source, since it is a much less studied field. Also, the techniques of identifying the source of images will be treated, since they have an important relationship. This work proposes a convolutional neuronal network capable of identifying the mobile devices that source a video from its frames.

8.4. Related Works

The tasks of forensic analysis of digital images can be divided into four branches, depending on the pursued objective [CFGL08]:

- Integrity verification.
- Recovery of the processing history.
- Classification based on the source.
- Grouping by source device and identification of the source.

Within the four groups, the *modus operandi* that is used is very similar: with the aim of implementing an algorithm that helps the analysis or to develop a new method that facilitates the classification, the scientific community studies the characteristics that make up or contains a digital content, to be able to use the information of the same ones in its investigations. These characteristics, which may be of a very different nature, such as the peculiarities of the camera sensor or the distortion of the lens, are the cornerstone of forensic analysis. In [VLCEK07, TNC10] a study is carried out on what can be the characteristics of an image that can be interesting or useful for the forensic analysis in mobile devices.

In the available literature, it is possible to several works based on the use of one or more of these characteristics. The main ones of them are [BBBT16, TCC16, FONBCS18, BLG+17, MG18, YHW12]. Their common point and the reason for being especially important for this work is the use by almost all of them a convolutional neuronal network for the identification of the source of acquisition, being the same technique that will be used in this work. [YHW12] does not use a CNN, but it carries out an investigation into the source of acquisition of a video, like this work, so it has been interesting to add its conclusions.

It should be noted that the literature available on the analysis of the identification of sources of acquisition is very unequal according to whether they deal with images or videos. Based on the first ones, there is an acceptable number of publications, although not very numerous. In contrast, the number of jobs that focus on the identification of the source of acquisition of a video is so much lower.

Therefore, encouraged by the results obtained in the literature when applying the techniques of Machine Learning to the images to classify them according to the camera that captured them, in this work we propose to apply those same techniques to videos. As a result, the proposal of this paper is the use of a network CNN for its classification.

This choice is due to being a technique very little studied in videos yet, and that according to the results of the most current literature, is one of the fastest and most effective methods that exists today. In addition, it is a method capable of adapting to different purposes with ease.

8.5. Workplan

The development of this work has been carried out in three phases:

1. **Research** This first phase was carried out in a period of approximately three months. It began with the choice of the topic of work together with the tutors: the identification of the source of acquisition of digital videos. Once chosen, those

three months were spent on information research. It was necessary to know what had already been done in order to advance and expand the already known results.

Thanks mainly to the subjects of Computational Geometry and Artificial Intelligence, it was possible to start with a small knowledge base on Machine Learning, Deep Learning and Neural Networks, which will be discussed in the next chapters. This knowledge was quickly extended thanks to the numerous articles of scientific journals, of congresses and some books on the subject, all of them found both in *Google Scholar* and in the libraries of the Complutense University of Madrid. It was searched mainly for publications on Automatic Learning, Neural Networks (especially convolutional ones), falsification of images and videos and identification of images and videos. Many of the works used in the research phase were found thanks to the references of other publications already studied.

2. **Development**: Once the research phase was sufficiently advanced and all the necessary knowledge had been acquired, the second phase was started, that of the development of the proposal. At this time it was determined what would be the proposal that would present this work: the development of a method that allows to identify the source of acquisition of a video from a CNN. At the same time that the proposal was developed and codified, research was also continued, although to a lesser extent than in the previous phase, only searching and consulting specific concepts. In this phase, the selection and creation of the sets of images and videos that would be used to train and test the proposed network was carried out, and the network was developed, codified, tested and redesigned on numerous occasions until the final design was achieved. To do this, it was investigated the programming language *Python* and its libraries *Keras, Tensorflow* and *Scikit Learn*.

3. **Results**: Finally, in the Results phase, it proceeded to the analysis of all the results that had been obtained with the numerous tests carried out and the extraction of conclusions from these results. This phase covered around two months and was overlapped a month and a half with the previous phase, since according to the results obtained, it was necessary to carry out more tests or not.

8.6. Struture of the Work

The rest of the work is organized in 9 chapters with the structure explained below:

Chapter 1 is an introduction of this work, which is the translation in spanish of this chapter.

Chapter 2 introduces the basic concepts of Machine Learning, Deep Learning and Neural Networks, which are essential to understand the foundations of the science on which the forensic analysis of images and videos is based.

Chapter 3 explains what is the identification of the source of acquisition of an image or video, what forensic techniques exist for it and what elements of the cameras are based to be applied. In this chapter we start by showing the different objectives pursued by forensic analysis: the determination of whether a content is original or not and the identification of the source camera. In both, a presentation of the methods and techniques used by all of them is made. Finally, it is shown in more detail what constitutes the identification of the source of acquisition from a CNN, since it is a main part of this work.

Chapter 4 contains a compilation of the most important investigations of the identification of the source of acquisition. In it, it can be read the achievements obtained by [BBBT16, TCC16] in the identification of the source camera of images captured by a

traditional camera DSC and how [FONBCS18] takes those results one step further applying those same techniques to images obtained from mobile devices. Next, the advances of [MG18, YHW12] are exposed in the field of videos, including those that have been transmitted through a social network.

Chapter 5 contains the proposal of this work. It explains in general what this proposal consists of and then details the final network that is proposed.

Chapter 6 describes the experiments carried out to evaluate the effectiveness of the convolutional neuronal network proposed in the chapter 5 and presents the results obtained.

Chapter 7 shows the main conclusions of this work and the possible future lines of research.

Chapters 8 and 9 are the English translations of the Introduction (Chapter 1) and the Conclusions (Chapter 7).

Capítulo 9

Conclusions and Future Work

In this last chapter it will be gathered the conclusions that have been reached after the development of the proposal of this work and the numerous experiments carried out. Then, possible future works that can continue this line of research, expand it and improve it will also be detailed.

9.1. Conclusions

Once this work is finished, the following conclusions have been reached:

In the literature currently available, there is a very large shortage in the investigation of the source of acquisition of the videos. As explained earlier in the introduction, this is a very big problem, since the rise of the techniques that are capable of manipulating videos makes it difficult to rely on the contents of the videos and their source of origin. This problem is being solved little by little with the images, but not with the videos, which take a much slower advance.

As it can be seen in the Experiments and Results chapter, an accuracy of 89 % has been obtained in the classification of videos according to the camera that recorded them with the convolutional neural network proposed in this work. These results, even being below 90 % and being inferior to the accuracy obtained working with images, can be considered very good. When affirming that they are good results, it has also been taken into account that, since there are very few publications that use a CNN to identify the source of acquisition of a video, this obtained precision is a more than decent and acceptable result as this is one of the first approaches to the identification of the source camera of a video with this type of classification method.

To carry out the experiments presented in this work, a personal computer with an 8GB RAM was used, which, a priori, seemed sufficient. However, it is necessary to highlight that due to the large number of images used and its high resolution, it soon became very short, slowing down the executions enormously (reaching around 3h each) or even becoming saturated and preventing its execution. This problem was mitigated in part with the use of the Google Collaboratory tool, which has up to 12GB of RAM. However, at the time of executing the experiments with Set 4, the shortage of memory became patent again, so it is suggested that an improvement in the resources used could represent a more agile execution and better results.

In this work, as explained in previous chapters, at the time of feeding the network, all the images are first divided into smaller areas or patches. This is so, for two reasons: the impossibility of executing the code with the images in their original resolution due to lack of memory, and the suggestions of [FONBCS18, BLG+17] to divide the images. We

want to point out that, contrary to what [FONBCS18] suggests, the best results have been obtained by dividing each image into pieces of 64x64 pixels, not 32x32 as they suggest. In addition, it has been necessary to raise the number of patches by 32 pictures proposed [BLG+17] to 256, obtaining the best accuracy with this figure.

The algorithm presented in this paper is designed in such a way that it presents great versatility, allowing it to perform a couple of extra tasks that are a little different from the original one:

First, its structure allows it to be able to analyze the source of image acquisition in addition to those of the videos. For that, it would be enough to comment on the code the part of the algorithm which is destined to the frames extraction and the final part in which the class of each video is discovered from the frames. It should also be ensured that the folder where the photographs to be analyzed are located has the same structure as the original folder *FRAMES*.

Secondly, the algorithm allows to analyze the precision of a previously trained and generated model, without the need to use the CNN proposed in the code and in this work.

9.2. Future Work

As possible future works following this investigation line, there are proposed the following ones:

- **Comparison of this work with a SVM.** A study on the identification of the video acquisition source using a SVM is proposed as a logical continuation of this work. It is thought that it may be a good idea to develop a SVM capable of identifying the camera that recorded a certain video from its frames, just as it has been done here, in order to be able to compare the results obtained with one and another classification method. This would allow to see which of the two obtains a higher precision, if it is possible to improve its execution time or if it could be carried out with less demanding resources than the convolutional neural network needs. The results that were thrown from here could determine which is the most suitable method to classify videos, and thus facilitate the increase of studies and results in this regard, since it is a field little studied.

- **Use of a SHAP or LIME type system.** Being able to explain a model is a priority nowadays for the scientific community, since it would avoid model biases and allow decision-makers to understand how the CNN work to use them in the best possible way. SHAP and LIME are two libraries of the programming language *Python* whose objective is to try to explain the model of a CNN. SHAP is responsible for considering all possible predictions of an instance from all its possible input values, and thanks to this comprehensive approach, it can guarantee properties such as consistency and local precision. LIME constructs scattered linear models of each prediction to explain its operation. The incorporation of these libraries to this work could provide interesting and useful information in order to improve the results presented and increase the accuracy already obtained.

- **Incorporation of the audio of the video.** In this work, to identify the source of acquisition of a video, only the information contained in its frames, previously extracted, is used. This means two things:

  Firstly, it means that all the information that the audio can contain is lost. When the audio is discarded, it is not taken into account if it provides valuable information

that can help to improve the classification of the video or, conversely, it may harm that classification. The incorporation of audio into the classification is proposed as future work, and the comparison of these results with those presented here.

The second meaning is that it may be that in a video the only element that is falsified is its audio. If this is the case, the study of its source of acquisition through only the frames is useless, since it will not detect any modification. It is therefore necessary to develop an additional study in which the source of the video's acquisition is evaluated from its audio, or added to this information. In this way, the work will be complemented and it is possible that the error rate of this work will be reduced.

Bibliografía

[ABHAN+19] H. Al Banna, A. Haider, J. Al Nahian, M. Islam, K.A. Taher, and S. Kaiser. Camera Model Identification using Deep CNN and Transfer Learning Approach. In *Proceedings of the International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST)*, Dhaka, Bangladesh, January 2019.

[Agg18] C.C. Aggarwal. *Neural Networks and Deep Learning.* Springer, 2018.

[BBBT16] L. Baroffio, L. Bondi, P. Bestagini, and S. Tubaro. Camera Identification with Deep Convolutional Networks. *CoRR*, March 2016.

[BFM+12] P. Bestagini, M. Fontani, S. Milani, M. Barni, A. Piva, M. Tagliasacchi, and S. Tubaro. An Overview on Video Forensics. *APSIPA Transactions on Signal and Information Processing*, 1:1229–1233, August 2012.

[BLG+17] L. Bondi, S. Lameri, D. Güera, P. Bestagini, E.J. Delp, and S. Tubaro. Tampering detection and localization through clustering of camera-based CNN features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Honolulu, HI, USA, July 2017.

[BM13] G.K. Birajdar and V.H. Mankar. Digital Image Forgery Detection Using Passive Techniques: A Survey. *ELSEVIER Digital Investigation*, 10(3):226–245, October 2013.

[Buc05] B.G. Buchanan. A (Very) Brief History of Artificial Intelligence. *AI Magazine*, 26(4):53–60, December 2005.

[CFGL08] M. Chen, J. Fridrich, M. Goljan, and J. Lukas. Determining Image Origin and Integrity Using Sensor Noise. *IEEE Transactions on Information Forensics and Security*, 3(1):74–90, March 2008.

[CLW06] K.S. Choi, E.Y. Lam, and K.K.Y. Wong. Automatic Source Camera Identification Using the Intrinsic Lens Radial Distortion. *Optics Express*, 14(24):11551–11565, November 2006.

[CPN+17] G. Cattaneo, U.F. Petrillo, M. Nappi, F. Narducci, and G. Roscigno. An Efficient Implementation of the Algorithm by Lukas et al. on Hadoop. In *Proceedings of the 12th International Conference on Green, Pervasive, and Cloud Computing*, pages 475–489, Cetara, Italy, April 2017.

[CR10] H.R. Chennamma and L. Rangarajan. Source Camera Identification Based on Sensor Readout Noise. *International Journal of Digital Crime and Forensics (IJDCF)*, 2(3):28–42, September 2010.

[CSA+18] D. Coombes, J. Sharp, G. Allix, J. Stirrup, and A. Moll. *Microsoft Official Course, 20774A Perform Cloud Data Science with Azure Machine Learning.* Microsoft, October 2018.

[FFG08] T. Filler, J. Fridrich, and M. Goljan. Using Sensor Pattern Noise for Camera Model Identification. In *Proceedings of the 2008 15th IEEE International Conference on Image Processing*, San Diego, CA, USA, October 2008.

[FONBCS18] D. Freire-Obregón, F. Narducci, S. Barra, and M. Castrillón-Santana. Deep Learning for Source Camera Identification on Mobile Devices. *Pattern Recognition Letters*, January 2018.

[GB11] T. Gloe and R. Böhme. The 'Dresden Image Database' for Benchmarking Digital Image Forensics. *Journal of Digital Forensic Practice*, 3(2–4):150–159, February 2011.

[GKWB07] T. Gloe, M. Kirchner, A. Winkler, and R. Bohme. Can We Trust Digital Image Forensics? In *Proceedings of the 15th International Conference on Multimedia*, pages 78–86, Augsburg, Germany, September 2007.

[GZY+18] D. Güera, F. Zhu, S.K. Yarlagadda, S. Tubaro, P. Bestagini, and E.J. Delp. Reliability Map Estimation for CNN-Based Camera Model Attribution. In *Proceedings of the 2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, Lake Tahoe, NV, USA, March 2018.

[JM15] M.I. Jordan and T.M. Mitchell. Machine Learning: Trends, Perspectives and Prospects. *Science*, 349(6245):255–260, July 2015.

[JMM96] A.K. Jain, J. Mao, and K.M. Mohiuddin. Artificial Neural Networks: a Tutorial. *Computer*, 29(3):31–44, March 1996.

[KAS17] S. Kingra, N. Aggarwal, and R.D. Singh. Video Inter-frame Forgery Detection Approach for Surveillance and Mobile Recorded Videos. *International Journal of Electrical and Computer Engineering (IJECE)*, 7(2):831–841, April 2017.

[KG15] M. Kirchner and T. Gloe. *Handbook of Digital Forensics of Multimedia Data and Devices*. John Wiley Sons, Ltd., July 2015.

[KSH12] A. Krizhevsky, I. Sutskever, and G.E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems*, pages 1097–1105, Lake Tahoe, NV, USA, December 2012.

[KSM04] M. Kharrazi, H.T. Sencar, and N. Memon. Blind Source Camera Identification. In *Proceedings of the 2004 International Conference on Image Processing, 2004. ICIP '04.*, pages 709–712, Singapore, Singapore, October 2004.

[Kub17] M. Kubat. *An Introduction to Machine Learning*. Springer, 2017.

[LBH15] Y. LeCun, Y Bengio, and G. Hinton. Deep Learning. *Nature*, 521:436–444, May 2015.

[LFG06] J. Lukas, J. Fridrich, and M. Goljan. Digital Camera Identification from Sensor Pattern Noise. *IEEE Transactions on Information Forensics and Security*, 1(2):205–214, June 2006.

[Li10] C.T. Li. Source Camera Identification Using Enhanced Sensor Pattern Noise. *IEEE Transactions on Information Forensics and Security*, 5(2):280–287, June 2010.

[LWY15] B. Liu, X. Wei, and J. Yan. Enhancing Sensor Pattern Noise for Source Camera Identification: An Empirical Evaluation. In *Proceedings of the 3rd ACM Workshop on Information Hiding and Multimedia Security*, pages 85–90, Portland, OR, USA, June 2015.

[MA13] M. Mishra and M.C. Adhikar. Digital Image Tamper Detection Techniques - A Comprehensive Study. *International Journal of Computer Science and Business Informatics*, 2(1):1–12, June 2013.

[MAU15] K. Mushtaque, K. Ahsan, and A. Umer. Digital Forensic Investigation Models: an Evolution Study. *JISTEM - Journal of Information Systems and Technology Management*, 12(2), August 2015.

[McC04] P. McCorduck. *Machines Who Think - A Personal Inquiry into the History and Prospects of Artificial Intelligence*. CRC Press, March 2004.

[MG18] C. Meij and Z. Geradts. Source Camera Identification Using Photo Response Non-Uniformity on WhatsApp. *ELSEVIER Digital Investigation*, 24:142–154, March 2018.

[RM15] S. Rachska and V. Mirjalili. *Python Machine Learning. Machine Learning and Deep Learning with Python, scikit-learn and TensorFlow*. Pckt Publishing Ltd., September 2015.

[RTD11] J.A. Redi, W. Taktak, and J.L. Dugelay. Digital Image Forensics: a Booklet for Beginners. *Multimedia Tools and Applications*, 51(1):133–162, January 2011.

[SAR+13] A.L. Sandoval, D.M. Arenas, J. Rosales, L.J. Garcia, and J.C. Hernandez-Castro. Techniques for Source Camera Identification. In *Proceedings of the 6th International Conference on Information Technology*, pages 1–9, Dubai, United Arab Emirates, May 2013.

[SM08] H.T. Sencar and N. Memon. Overview of State-of-the-Art in Digital Image Forensics. *Algorithms, Architectures and Information Systems Security*, pages 1325–347, November 2008.

[SXD09] Y. Su, J. Xu, and B. Dong. A Source Video Identification Algorithm Based on Motion Vectors. In *Proceedings of the Second International Workshop on Computer Science and Engineering*, pages 312–316, Qingdao, China, October 2009.

[TCC16] A. Tuama, F. Comby, and M. Chaumont. Camera Model Identification with the Use of Deep Convolutional Neural Networks. In *Proceedings of the 2016 IEEE International Workshop on Information Forensics and Security (WIFS)*, Abu Dhabi, United Arab Emirates, December 2016.

[TNC10] V. L. L. Thing, K. Y. Ng, and E. C. Chang. Live Memory Forensics of Mobile Phones. *Digital Investigation*, 7:74–82, August 2010.

[VLCEK07] T. Van Lanh, K. S. Chong, S. Emmanuel, and M. S. Kankanhalli. A Survey on Digital Camera Image Forensic Methods. In *Proceedings of the IEEE International Conference on Multimedia and Expo*, pages 16–19, Beijing,China, July 2007.

[YHW12] S. Yahaya, A.T.S. Ho, and A.A. Wahab. Advanced Video Camera Identification Using Conditional Probability Features. In *Proceedings of the IET Conference on Image Processing*, pages 1–5, London, United Kingdom, July 2012.

[YNZZ19] P. Yang, R. Ni, Y. Zhao, and W. Zhao. Source Camera Identification Based on Content-Adaptive Fusion Residual Networks. *ELSEVIER Pattern Recognition Letters*, 119:195–204, March 2019.

[ZZC+17] K. Zhang, W. Zuo, Y. Chen, L. Zhang, and D. Meng. Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, July 2017.